# Objektorientierte Planspielentwicklung

## - dargestellt am Beispiel eines Versicherungsplanspiels

Inaugural-Dissertation
zur Erlangung des Grades oeconomiae publicae
an der Ludwig-Maximilians-Universität München

vorgelegt von: Ralf Klotzbücher

1995

Referent: Prof. Dr. E Helten
Korreferent: Prof. Dr. A. Picot

Promotionsabschlußberatung: 26. Juli 1995

# 1. Vorwort des Herausgebers

Die Ausgangsposition ist klassisch: Die quantitative Betriebswirtschaftlehre bringt die Vorstellung von Unternehmen als mathematisch zu beschreibende Systeme, die allgemeine Systemtheorie legt die Grundlage, um Unternehmen und Märkte als vernetzte Systeme zu erklären und die Risikotheorie entwickelt das mathematisch fundierte Konzept für die Versicherungswissenschaft.

Der Wandel als Kontinuum, Dynamik durch Innovation, veränderte Rahmenbedingungen und die immer weiter gestiegene Komplexität machen es gerade für die Aus- und Weiterbildung nötig, sich neuer Konzepte und Techniken zu bedienen - Unternehmensplanspiele können gerade hier einen wertvollen Beitrag leisten.

Vor dem Hintergrund der Liberalisierung des Europäischen Marktes ist es gerade für die Aus- und Weiterbildung in der Versicherungswirtschaft besonders wichtig, sich diesen Herausforderungen zu stellen.

Die vorliegende Arbeit leistet eine Verknüpfung der Disziplinen zur Gestaltung von Unternehmensplanspielen und stellt dies am Beispiel eines Versicherungsplanspiels dar:

> Die durch die Vorstellungen des Konstruktivismus maßgebliche beeinflußte Reformpädagogik fordert handlungsorientierte, lernerzentrierte Ausbildung in authentischen Situationen. Es zeigt sich, daß Planspiele gleichmaßen Werkzeug und Methode zur Verwirklichung dieser Vorstellungen sein kann.
>
> Die Informatik gibt mit der objektorientierten Methodik einen allgemein verständlichen Rahmen für Modellbildung und -implementierung, an der alle Beteiligten aktiv mitwirken können.
>
> Langjährige praktische Erfahrung mit dem Instrument Planspiel und kreative Abbildung eines durch Risikotheorie, Versichrungsbetriebslehre und Organisations-psychologie fundierten Komponentenmodells ermöglichen einen Baukasten zur **lernzielorientierten** Gestaltung von Unternehmensplanspielen - eine wiederverwend-bare und erweiterbare Objektbibliothek entsteht.

Letztendlich bringt die Technik den Menschen wieder in den Mittelpunkt des Interesses.

München, September 1995

Elmar Helten

# 1. Inhaltsverzeichnis

Vorwort des Herausgebers ..... V

Inhaltsverzeichnis ..... VII

Abkürzungsverzeichnis ..... XI

Abbildungsverzeichnis ..... XIII

Tabellenverzeichnis ..... XV

1 Einführung ..... 1

1.1 Die Idee ..... 1

1.2 Der rote Faden ..... 7

2 Grundlagen zu Planspielen ..... 10

2.1 Systeme ..... 10

2.2 Simulation ..... 14

2.3 Planspiele und Planspielentwicklung ..... 23

3 Anforderungsorientierte Planspiele ..... 31

3.1 Planspiele als Instrument der Aus- und Weiterbildung ..... 31

3.1.1 Lernen mit Planspielen ..... 32

3.1.2 Planspiele als Instrument der Aus- und Weiterbildung an Schulen und Universitäten ..... 42

3.1.3 Planspieleinsatz in der betrieblichen Aus- und Weiterbildung für Führungskräfte ..... 47

3.2 Entwicklung eines Kriterienrahmens für anforderungsorientierte Planspielentwicklung ..... 48

3.2.1 Anforderungen an Planspiele ..... 48

3.2.2 Anforderungen an die Implementierung von Planspielen ..... 53

4 Einsatz des objektorientierten Paradigmas zur Entwicklung von Unternehmensplanspielen ..... 59

4.1 Objektorientierte Systeme ..... 59

4.1.1 Objektorientierte Modellierung ..... 59

4.1.2 Objektorientierte Programmierung ..... 64

4.1.3 Objektorientierte Datenbanken ..... 74

4.2 Möglichkeiten des objektorientierten Paradigmas für die Entwicklung von Planspielen ..... 78

4.2.1 Ansätze zur Entwicklung von Planspielen ..... 78

4.2.2 Modellbildung und Implementierung bei objektorientierter Planspielentwicklung ..... 85

4.3 Bewältigung der Problemdimensionen der Planspielentwicklung ..... 90

4.3.1 Model-View-Controller - das grundlegende Gliederungsprinzip ..... 91

4.3.2 Statik und Dynamik - zwei Sichtweisen der Modellierung ..... 95

4.3.3 Mikro oder Makro - die entscheidende Frage für die Modellbildung ..... 97

4.3.4 Komplexität des Modells - Möglichkeiten zur Variation ..... 101

4.4 Ein qualitätsorientiertes Konzept für die objektorientierte Entwicklung von Planspielen ..... 105

4.4.1 Die Position der Qualitätsorientierung .....105

4.4.2 Ein objektorientiertes Vorgehensmodell der Planspielentwicklung .....111

4.4.3 Neue Möglichkeiten der Organisation der Planspielentwicklung .....113

5 Objektorientierte Entwicklung von Unternehmensplanspielen mit Object-VersPlan ..... 118

5.1 Eine Objektbibliothek für ein Versicherungsplanspiel.....119

5.1.1 Klassenhierarchie .....119

5.1.2 Simulation mit Object-VersPlan.....126

5.1.3 Metamodell.....128

5.2 Werkzeuge zur Entwicklung und Durchführung von Unternehmensplanspielen.....129

5.2.1 Zielgruppen und Werkzeuge .....130

5.2.2 Konzepte.....131

5.2.3 Beschreibung der Werkzeuge.....133

5.2.4 Systemarchitektur .....144

5.3 Die Anwendung von Object-VersPlan - dargestellt an ausgewählten Beispielen .....146

5.3.1 Aufbauorganisation .....146

5.3.2 Ablauforganisation .....156

6 Zusammenfassende Bewertung und Ausblick ..... 166

6.1 Eignung des objektorientierten Ansatzes zur Planspielentwicklung .... 166

6.2 Bezüge objektorientierter Planspielentwicklung zu anderen Problemklassen ..... 169

6.3 Ausblick ..... 172

Literaturverzeichnis ..... 175

Autorenverzeichnis ..... 187

Stichwortverzeichnis ..... 190

# 2. Abkürzungsverzeichnis

| | |
|---|---|
| B/E-System | Bedingung/Erreichbarkeits-System (Petrinetz-Theorie) |
| bzw. | beziehungsweise |
| CASE | Computer Aided Software Engineering |
| CBT | Computer Based Training |
| CORBA | Common Object Request Broker Architecture |
| CRC | Class Responsibilities Collaboration |
| ders. | derselben |
| d. h. | das heißt |
| DSOM | Distributed System Object Model |
| EDV | Elektronische Datenverarbeitung |
| engl. | im Englischen |
| etc. | et cetera |
| f. | folgende |
| ff. | fortfolgendende |
| Hrsg. | Herausgeber |
| INRIVER | Institut für Betriebswirtschaftliche Risikoforschung und Versicherungswirtschaft (Prof. Dr. E. Helten, Universität München) |
| K/I-Netz | Kanal/Instanzen-Netz (Petrinetz-Therorie) |
| M-V-C | Model-View-Controller |
| OBA | Object Behaviour Analysis |
| ODMG | Object Database Management Group |
| OMG | Object Management Group |
| OMA | Object Management Architecture |
| OMT | Object Modelling Technique |
| OOA | Objektorientierte Analyse |
| OOD | Objektorientiertes Design |
| OODBMS | Objektorientiertes Datenbank Management System |
| OOP | Objektorientierte Programmierung |

| | |
|---|---|
| PC | Personal Computer |
| Pr/T-Netz | Prädikat/Transitionen-Netz (Petrinetz-Therorie) |
| RDD | Responsibility Driven Design |
| S. | Seite |
| SOM | System Object Model |
| Sp. | Spalte |
| SQL | Structured Query Language |
| S-V-C | Subject-View-Controller |
| S/T-Netz | Stellen/Transitionen-Netz (Petrinetz-Therorie) |
| TQM | Total Quality Managements |
| VB | Visual Basic |
| vgl. | vergleiche |
| vgl. a. | vergleiche auch |
| VW | Versicherungswirtschaft |
| WWW | World Wide Web |
| z.B. | zum Beispiel |
| ZfB | Zeitschrift für Betriebswirtschaft |
| zfbf | Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung |
| ZVersWiss | Zeitschrift für die Gesamte Versicherungswissenschaft |

# 3. Abbildungsverzeichnis


Abbildung 1: Repräsentation des Kontextes in Planspielen ..... 24
Abbildung 2: Spielsituation ..... 25
Abbildung 3: Die Problemfelder der Planspielentwicklung ..... 27
Abbildung 4: Kapselung ..... 65
Abbildung 5: Nachricht ..... 66
Abbildung 6: Klassenhierarchie ..... 67
Abbildung 7: Vererbung ..... 68
Abbildung 8: Schichtenmodell der Subjekt-View-Controller Architektur ..... 92
Abbildung 9: Vertikale und horizontale Komplexitätsvariation ..... 101
Abbildung 10: Kunden und Lieferanten der Planspielentwicklung ..... 107
Abbildung 11: Vorgehensmodell der Planspielentwicklung ..... 111
Abbildung 12: Teilehierarchie ..... 122
Abbildung 13: Simulationsverbindungen ..... 127
Abbildung 14: Metamodell ..... 128
Abbildung 15: Werkzeuge und M-V-C ..... 132
Abbildung 16: Ausschnitt aus der Klassenhierarchie für die Controller ..... 132
Abbildung 17: Metamodell aus Sicht des Teile-Editors ..... 134
Abbildung 18: Teile-Editor ..... 135
Abbildung 19: Metamodell aus Sicht des Ablauf-Editors ..... 136
Abbildung 20: Ablauf-Editor ..... 137
Abbildung 21: Entscheidungen-Manager ..... 137
Abbildung 22: Eingabedialog für Entscheidungen ..... 138
Abbildung 23: Metamodell aus Sicht der Simulatorwerkzeuge ..... 139
Abbildung 24: Aufgabenteilung während der Simulation ..... 140
Abbildung 25: Simulator ..... 140
Abbildung 26: Debugger ..... 141
Abbildung 27: Metamodell aus Sicht des Berichte-Editors ..... 142

Abbildung 28: Bericht ..... 143
Abbildung 29: Systemarchitektur von Object-VersPlan ..... 144
Abbildung 30: Aufbauorganisation eines Planspiels ..... 146
Abbildung 31: Unternehmen ..... 147
Abbildung 32: Globale Entscheidungen zum Unternehmen ..... 148
Abbildung 33: Modellierung eines Kollektivs von Gewerberisiken ..... 150
Abbildung 34: Diskrete Schadenzahlverteilung ..... 151
Abbildung 35: Abteilung ..... 152
Abbildung 36: Mitarbeiter ..... 153
Abbildung 37: Motivation in Abhängigkeit von der Bezahlung ..... 154
Abbildung 38: Schadenbearbeitung ..... 155
Abbildung 39: Ablauf einer Periode ..... 157
Abbildung 40: Schadensimulation ..... 158
Abbildung 41: Ablauf der Marktsimulation ..... 160
Abbildung 42: Auktionator ..... 161
Abbildung 43: Modellierung der Markteffekte ..... 161
Abbildung 44: Kennzahlen des Auktionators ..... 162
Abbildung 45: Marktsimulation: Angebot ..... 163
Abbildung 46: Verwaltung (VU 1) ..... 163

# 1. Tabellenverzeichnis


Tabelle 1: Planspielentwicklung im interdisziplinären Kontext ....8

Tabelle 2: Strukturkonzepte für Systeme ....11

Tabelle 3: Abgrenzung von Simulationsverfahren ....19

Tabelle 4: Diskrete Simulation ....21

Tabelle 5: Zielkonflikte bei der Gestaltung von Lernumgebungen ....36

Tabelle 6: Lehrmethoden des Cognitive Apprenticeship-Ansatzes ....37

Tabelle 7: Forderungen aus dem Kontext der Aufgabe an Planspiele ....38

Tabelle 8: Forderungen übergeordneter Lernziele moderner Lerntheorien an Planspiele ....39

Tabelle 9: Forderungen der Lehr/Lernumgebung moderner Lerntheorien an Planspiele ....40

Tabelle 10: Lernsituationen für Planspiele ....41

Tabelle 11: Die Mikrowelt im Planspiel-Simulationsmodell ....50

Tabelle 12: Unterstützung verschiedener Lehr-/Lernsituationen ....51

Tabelle 13: Dokumentation des Modells ....52

Tabelle 14: Lernsituationen und deren Anforderungen an die Planspielentwicklung ....57

Tabelle 15: Objektorientierte Analyse und Objektorientiertes Design ....62

Tabelle 16: Kriterien zur Beurteilung von Ansätzen zur Planspielentwicklung ....78

Tabelle 17: Transformation der realen Welt im objektorientierten Ansatz ....86

Tabelle 18: Modelltypologie von Absatzmodellen ....99

Tabelle 19: Klassenhierarchie der Modell-Klassen ....121

Tabelle 20: Klassenhierarchie der View-Klassen ....124

Tabelle 21: Klassenhierarchie der Controller-Klassen ....125

Tabelle 22: Zielgruppen ....130

Tabelle 23: Zielgruppen, Aufgaben und Werkzeuge ....130

Tabelle 24: Bezüge von Konzepten objektorientierter Planspiele zu anderen Anwendungsfällen ....169

# 1. Einführung

„Fehler sind erwünscht“[1],

so konnte man in einem deutschen Wirtschaftsmagazin als Überschrift lesen: Mit Unternehmensplanspielen steht der Aus- und Weiterbildung eine mächtige Methode zur Verfügung. Planspiele können mit Hilfe realistischer Computersimulationen eine Lernumgebung schaffen, die Fach-, System- und Sozialkompetenz besser vermitteln kann als andere Lernformen.

Ein computergestütztes Simulationsmodell ist für ein Planspiel in vielen Fällen die notwendige Basis - und zugleich aber doch nur Mittel zum Zweck. In der vorliegenden Arbeit wird deshalb versucht, die Entwicklung eines anforderungsorientierten Planspiels bis hin zur Implementierung als Versicherungsplanspiel zu verfolgen.

Dabei werden die grundlegenden Ideen und Konzepte der objektorientierten Methodik durchgängig und konsistent in allen Phasen angewendet und der Weg zu einem in der Praxis anwendbaren Planspielprogramm aufgezeigt.

## 1.1. Die Idee

Wie es zum Thema der vorliegenden Arbeit gekommen ist, soll im folgenden gezeigt werden. Der anschließende Überblick über die zentralen Ideen hat Vorschaucharakter auf die später noch ausführlich zu diskutierenden Aspekte.

### Motivation

Am **INRIVER** der Universität München[2] wurde in den Jahren 1988 bis 1991 *Hansens* Integriertes Versicherungsplanspiel[3] vom Großrechner auf PC umgesetzt. Dieses Planspiel wird nun seit einigen Jahren im Rahmen eines Seminars für die Ausbildung von Studenten mit Studienschwerpunkt Versicherungen und in Kooperation mit Partnern aus der Versicherungswirtschaft als Weiterbildungsinstrument für Führungskräfte und Führungskräftenachwuchs eingesetzt. Die Erfahrungen bei Entwicklung und Einsatz dieses computerunterstützten Unternehmensplanspiels haben den entscheidenden Anstoß

1 Schwertfeger, B. (Fehler, 1992), S. 68

2 INRIVER ist das Institut für Betriebswirtschaftliche Risikoforschung und Versicherungswirtschaft, Prof. Dr. E. Helten, an der Universität München.

3 Vgl. Werner, U. (VersPlan, 1991), S. 1ff.

gegeben, sich fundiert mit Entwurf und Implementierung eines von Grund auf neu konzipierten „Planspiels Versicherungen“ zu beschäftigen.

Die Lehrpraxis hat gezeigt, daß das Planspiel ein mächtiges Instrument zur Ausgestaltung von Lehr-/Lernsituationen sein kann. Es bietet eine mögliche Antwort auf die Forderungen der **Reformpädagogik** nach konstruktivistisch orientierten Lernansätzen. Was liegt also näher, als sich intensiver mit den Gestaltungsmöglichkeiten für Planspiele auseinanderzusetzen?

Ein weiterer Anstoß kam von Seiten der Technik: Beschränkungen und Nachteile konventioneller Modellbildung und Implementierung von **Simulationsmodellen** haben großen Einfluß auf die Einsatzmöglichkeiten eines Planspiels in der Aus- und Weiterbildung: Technische Aspekte diktieren den Einsatz - und nicht umgekehrt. In der Folge soll versucht werden, Ansätze zu entwickeln, die diese Grenzen mildern oder ganz aufheben.

Der objektorientierte Ansatz setzt sich nach mehr als zwanzig Jahren Entwicklung und Reife in Hochschulen, Forschungseinrichtungen und Nischenanwendungen mittlerweile auf breiter Basis auch in kommerziellen Projekten der Individualsoftwareentwicklung und bei der Entwicklung von Standardapplikationen durch:[4] Objektorientierung ist dabei methodische Basis von Entwurf und Modellierung sowie zugleich Paradigma der Implementierungstechnik. Gerade die Entwicklungen im Bereich der Bürokommunikation, der Groupware[5], der Unternehmensmodellierung und der Modellierung von Informationssystemen[6] können Anregungen für die objektorientierte Entwicklung von Planspielen liefern.

Angeregt durch die Arbeit *Frickers*[7], der die **systemtheoretisch** fundierte Sichtweise eines Versicherungsunternehmens zum Entwurf eines Führungsmodells genutzt hat, reifte die Idee, bei Modellierung und Implementierung eines Planspiels der systemorientierten Denkweise zu folgen. Die Prinzipien Systemorientierung und Objektorientierung ergänzen sich - sie sollen in allen Phasen der Planspielentwicklung als Grundlage verwendet werden.

Ergänzend zu diesen Erfahrungen und Anregungen kommt die Beschäftigung des Autors mit objektorientierter Softwareentwicklung während einer studienbegleitenden Tätigkeit und im Rahmen der Diplomarbeit[8] hinzu. Diese wichtige Grundlage erst macht es möglich, sich auch praktisch mit dem Thema auseinanderzusetzen - denn objektorientierte Softwareentwicklung ist nur in langjähriger Praxis an konkreten Projekten zu erlernen und effektiv einzusetzen.

---

4 Vgl. Skudelny, H. (Objektorientierung, 1995), S. 62 f. ; vgl. a. Wagner, M. (Objektorientierung, 1995), S. 6

5 siehe Lautenschläger, S. (Groupware, 1994)

6 siehe Frank, U. , Klein, S. (Tools, 1992); siehe auch ders. (Unternehmensmodelle, 1992)

7 siehe Fricker, U. (System, 1982)

8 siehe Klotzbücher, R. (Aspekte, 1990)

## „Aus dem Fach heraus“ [9]

Die vorliegende Arbeit versteht sich im Sinne der Ideen *Vesters* als eine Arbeit aus dem Fach heraus: Es soll insbesondere versucht werden, eine Aufgabe aus dem Bereich der Betriebswirtschaftslehre unter Einbeziehung konvergierender Gesichtspunkte von Forschungsaktivitäten anderer Disziplinen zu lösen:

Objektorientierte Planspielentwicklung bewegt sich im Schnittbereich der Disziplinen angewandte Informatik, Mathematik und Statistik, Wirtschaftswissenschaften und Pädagogik - interdsziplinäres Denken, wie es der systemorientierten Denkweise entspricht[10].

Damit die Kommunikation der Ideen und Lösungsansätze auch fachübergreifend möglich ist, wird auf eine verständliche und weitestgehend von Fachjargon befreite Darstellung Wert gelegt.

Durch die praktische Umsetzung der hier vorgestellten Ideen als Prototyp eines lauffähigen Computerprogramms soll deren Propagierung weiter unterstützt werden.

## Objektorientiert zu anwendungsbestimmten Planspielen

In einem Planspiel erhalten die Teilnehmer die Möglichkeit, in einer Spielsituation in die modellierten Zusammenhänge der Realität einzutauchen. Kern jedes Planspiels ist ein Simulationsmodell - ein vereinfachtes Abbild eines relevanten Ausschnittes der Realität.

## Objektorientiertes Paradigma und Systemansatz

> Was liegt näher, als sich sich bei der Erstellung dieses Simulationsmodells einer Methodik zu bedienen, die sich sehr nahe an diesem Wirklichkeitsausschnitt orientiert?
>
> Aus diesem Grunde sollen die Möglichkeiten des **objektorientierten Paradigmas** untersucht werden:
>
> Im objektorientierten Verständnis wird die Realität wird als Gesamtheit von Objekten mit Eigenschaften und Verhalten aufgefaßt. Diese Objekte stehen - wie auch in der Wirklichkeit - zueinander in Beziehungen; sie lassen sich klassifizieren und einordnen und sie unterhalten miteinander Kommunikationsbeziehungen.
>
> Diese Sichtweise steht dem systemorientierten Ansatz sehr nahe, der Ausschnitte aus der Realität als System von Elementen und deren Verknüpfungen versteht. Objektorientierte Systeme modellieren also nicht die Realität, sondern mithin die Vorstellung von einem Systemmodell.

9 Vester, F. (Neuland, 1991), S. 482

10 siehe De Rosnay, J. (Makroskop, 1977)

Beim Einsatz des objektorientierten Ansatzes in der Praxis stehen Wirtschaftlichkeits- und Qualitätsaspekte bei der Erfüllung der Anforderungen des Einsatzzwecks im Vordergrund. Deshalb wird gefordert, den Aspekten der Wiederverwendung vorgefertigter Komponenten, der Orientierung an den Anforderungen und Bedürfnissen der Zielgruppen und deren Unterstützung durch ein leistungsfähiges Instrumentarium von Werkzeugen verstärkte Beachtung zu schenken.

## Objekt-Bibliotheken und Wiederverwendung

Objekte sind diejenigen Elemente, aus denen ein Planspiel-Simulationsmodell besteht. Als Bausteine aufgefaßt, können einzelne Objekte oder auch Gruppen von Objekten[11] in **Bibliotheken** oder Datenbanken gehalten werden. Sie stehen als vorgefertigte, wiederverwendbare Bausteine bei der Konstruktion eines Planspiels zur Verfügung. Durch Erweiterung dieser Objektbibliothek[12] könnte der Einsatzbereich eines Planspiel-Entwicklungswerkzeuges ausgeweitet werden. Je nach Spezialisierungsgrad wären solche Objekte für verschiedene Anwendungsbereiche nutzbar: Die objektorientierte Methodik verspricht, eine anwendungsneutrale Basis zur Entwicklung von Planspielen für die unterschiedlichsten Anwendungsfälle sein zu können.

In der vorliegenden Arbeit soll die Anwendung für Unternehmensplanspiele aus dem Bereich der Versicherungswirtschaft im Vordergrund stehen.

## Anwendungsbestimmte Planspiele

Objektorientierung kann den Weg zur Realisierung von computerunterstützten Unternehmensplanspielen aufzeigen, die sich nicht in erster Linie von den Möglichkeiten der Technik, sondern von den Anforderungen der Einsatzsituation leiten lassen. Flexibilität und Klarheit des Ansatzes und semantische Nähe zum Problem können die Basis für individuelle Planspiele legen. Im Gegensatz zu traditionellen Planspielen, die in wesentlichen Aspekten von der Technik bestimmt werden, kann ein mit Hilfe des objektorientierten Ansatzes entwickleltes Planspiel genauer den spezifischen Anforderungen und Zielen des Anwendungsfalles - also zum Beispiel der **Seminarsituation** (ob Fernplanspiel, Blockseminar oder interaktive Selbstlern-

11 Ein einzelnes, wiederverwendbares Objekt könnte zum Beispiel eine Gruppe von Mitarbeitern im Innendienst eines Versicherungsunternehmens sein. Als Beispiel für eine Gruppe von Objekten könnte man sich ein ganzes Unternehmen vorstellen. Dieses wird als eine Hierarchie von Teilen (Bereiche, Abteilungen, Mitarbeiter etc.) abgebildet und kann als Ganzes wiederverwendet werden. So können in einem Unternehmensplanspiel Unternehmen, die man einer Objektdatenbank entnehmen kann, mit unterschiedlichen Organisationsformen auf einem simulierten Markt miteinander konkurrieren.

12 Häufigster Fall der Erweiterung ist die Ableitung neuer Objekte von bestehenden Objekten durch Spezialisierung: Es müssen nur diejenigen Eigenschaften und Fähigkeiten neu implementiert werden, die verändert werden oder die neu hinzukommen. Eine Erweiterung der Objektbibliothek kann auch durch Kombination bestehender Objekte geschehen.

umgebung) und den Vorkenntnissen und **Lernzielen** der Teilnehmer - angepaßt werden.

Objektorientierte Methodik und Technologie können eine fundierte Basis setzen, um diese Anforderungen mit vertretbarem Aufwand zu realisieren:

- Der spezifische Kontext des Anwendungsfalls findet sich in der Modellierung spezialisierter Objekte wieder[13].

- Die in zahlreichen Anwendungsfällen wiederkehrenden Bestandteile eines Planspiels (beispielsweise Teilnehmer, Spielleiter, Unternehmen, Kunden etc.) können in vielen Fällen unverändert oder mit angepaßten Eigenschaften übernommen werden. Sie bilden die Wurzeln in einer baumartigen Hierarchie wiederverwendbarer Objekte.

- Planspiele könnten mit Hilfe der objektorientierten Methodik ihre Stärken zur Unterstützung neuerer Methoden des Lernens sehr gut ausspielen: Aktives, vernetzes Lernen in authentischen Situationen, wie es von *Vester*[14] oder den konstruktivistisch orientierten Reformpädagogen[15] angeregt und gefordert wird, legt die theoretischen Grundlagen für die Gestaltung von **Lehr-/Lernsituationen.** Jene sind mithin Motivation für anforderungsorientierte Unternehmensplanspiele:

    1. Je nach Anforderungen der Lernziele kann die Komplexität des zugrundeliegenden Modells auch während des Spiels in weitem Maße verändert werden.

    2. Für den Lernerfolg weniger wichtige Problembereiche können mit gröberer Granularität modelliert werden, während die Lernschwerpunkte detailliert im Planspielmodell Nachbildung finden.

    3. Je nach Seminarsituation können das Verhalten der Objekte und die Möglichkeiten der Teilnehmer, selbst aktiv in das Spielgeschehen einzugreifen, angepaßt werden.

---

13 Im Anwendungsfall Versicherungsplanspiel könnte dies zum Beispiel das Modell eines zu versichernden Risikos oder eines Kollektivs von Risiken sein.

14 Vgl. Vester, F. (Neuland, 1991), S. 456 ff.; siehe auch Vester, F. (Denken, 1978)

15 Vgl. Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993), S. 9 ff.;
vgl. a. Collins, A., Brown, S., Newman, S. (Cognitive Apprenticeship, 1989), S. 453 ff.

## Bewältigung der Komplexität

Die Bewältigung zunehmend komplexer Systeme stellt eine Herausforderung für die Innovationskraft von Forschung und Entwicklung dar: Es sind geeignete, weiter entwickelte Technologien gefordert, mittels derer es gelingt, höhere Komplexität nicht mit größerer Kompliziertheit erkaufen zu müssen[16].

Für Planspiele kann man unter diesem Aspekt das Thema Komplexität aus verschiedenen Blickwinkeln betrachten:

- Bei der Erstellung muß es gelingen, die Komplexität des Anwendungsbereiches in der Wirklichkeit auf ein anforderungsgerechtes Maß im Modell zu reduzieren. Dabei ist darauf zu achten, daß die Modellierungstechnik das Problem nicht unnötig kompliziert. Die objektorientierte Vorgehensweise tritt an, die semantische Lücke von der Modellierung zum realen Problem schließen zu können; sie kommt dieser Forderung weitgehend entgegen.
  Durch das Zusammenfügen vorgefertigter, wiederverwendbarer Objekte variabler Komplexität und Detailliertheit kann dem Planspielentwickler unnötige Kompliziertheit erspart werden, ohne an der Komplexität der Modellierung Abstriche machen zu müssen.

- Im konstruktivistischen Verständnis einer Planspielsituation sollen Spielleiter und Teilnehmer die Möglichkeit haben, sich interaktiv, selbstgesteuert und schrittweise die Wirkzusammenhänge des Modells zu erarbeiten und Rückschlüsse auf die Tatbestände in der Realität zu ziehen: In einer anforderungsgerechten, objektorientierten Planspielumgebung können sie sich über die Beschäftigung mit einem Modell der Komplexität der Realität nähern - und das, ohne von Anfang an mit unnötiger **Kompliziertheit** belastet zu werden.

- Ein wesentliches Lernziel von Planspielen ist das Erkennen und Beherrschen von vernetzten, komplexen Systemen. Objektorientiert modellierte Planspiele präsentieren den Teilnehmern eine Lernsituation, die sich variabel an ihren Erfahrungen, ihrem Lernfortschritt und ihren Bedürfnissen orientiert.

## Leistungsfähige Werkzeuge

Ein wesentliches Hilfsmittel, um die Beherrschung der Komplexität möglich zu machen, sind leistungsfähige Werkzeuge. Sie unterstützen Planspielentwickler, Spielleiter und Teinehmer bei Entwicklung und Durchführung von Planspielen.

---

16 Vgl. Biedenkopf, K. H. (Komplexität, 1994), S. 82 ff.
BIEDENKOPF zeigt am Beispiel des Systems der öffentlichen Verwaltung wie durch Innovationen ohne Erhöhung der Kompliziertheit Probleme mit höherer Komplexität bewältigt werden können.

Der objektorientierten Methodik folgend, ist es möglich, ohne Wechsel des Paradigmas anforderungsgerechte Werkzeuge zu erstellen. Die Planspielkonstruktion kann beispielsweise mit graphischen Editoren für das Zusammenfügen der Elemente (im objektorientierten Ansatz: Objekte) des Simulationsmodells maßgeblich unterstützt werden. Ein Anwender (ein Spiel- oder Seminarleiter) mit Detailkenntnissen über den spezifischen Anwendungsbereich kann mit solchen Werkzeugen ohne tiefergehende Kenntnisse der Programmierung, Mathematik oder Informatik ein von den Anforderungen des Planspielkonzeptes bestimmtes Planspielmodell konstruieren.[17]

Die Dokumentation von Struktur, Zusammenhängen, Eingriffsmöglichkeiten und Wirkungen für die Benutzer (seien es Konstrukteur, Spielleiter oder Teilnehmer) ist eine wesentliche Voraussetzung, um Nutzen aus dem Modell ziehen zu können. Objektorientierte Werkzeuge könnten hierzu durch den Einsatz anwenderorientierter Visualisierungs- und Interaktionstechniken ein wichtigen Beitrag leisten.[18]

## 1.2. Der rote Faden

Wie können nun die oben ausgeführten Ideen in ein Konzept zur Entwicklung anforderungsgerechter Planspiele umgesetzt werden?

### Ziele

Die Ziele der vorliegenden Arbeit lassen sich in vier Punkten zusammenfassen:

1. Die Aufgabe Planspielentwicklung soll als interdisziplinäre Aufgabe begriffen, eingeordnet und diskutiert werden.

2. Es soll ein Kriterienrahmen in Form ein Anforderungsprofils an Planspiele erarbeitet werden. Dafür sollen Modellierung, Implementierung und Systemtechnik berücksichtigt werden.

---

17 Es soll nicht verschwiegen werden, daß in der Anwendung von Werkzeugen, die komplizierte Sachverhalte durch einfach zugängliche Funktionalität verdecken, auch die Gefahr der Fehlanwendung liegt. Dies könnte aber zum Beispiel durch anwendungsnahe Dokumentation und die Wiederverwendung ausgetesteter Teile sowie Gruppen von Teilen gemildert werden.

18 Durch die Trennung von logischer Ebene und Präsentations- und Interaktionsebene des Modells im Model-View-Controller Ansatz wird es möglich, ein und denselben Sachverhalt in unterschiedlicher Form an den Benutzer angepaßt zu präsentieren (z. B. als Gleichung, als Tabelle oder als Grafik). Gleiches gilt für die Abstufung der Interaktionsmöglichkeiten mit dem Modell: Wo für erfahrene Benutzer die Eingabe von genauen Werten passend ist, mag für andere die interaktive Bearbeitung von Grafiken richtig sein.

3. Es sollen die Möglichkeiten und Grenzen der objektorientierten Methodik für die Planspielentwicklung herausgearbeitet werden.

4. Am Beispiel eines im Rahmen der vorliegenden Arbeit entstandenen Prototyps eines Versicherungsplanspiels soll gezeigt werden, wie Konzepte zur anforderungsgerechten Planspielkonstruktion in die Praxis umgesetzt werden können.

Parallel zu den theoretischen Arbeiten ist deshalb das Computerprogramm „Object-VersPlan 2.0“[19] entstanden: Es wurden mächtige Werkzeuge zur Konstruktion und Durchführung von Planspielen und der Grundstock einer auf Versicherungsplanspiele spezialisierten Objektbibliothek realisiert.

An diesem Beispiel sollen Aspekte aufgezeigt werden, wie das Ziel anforderungsorientierter Planspielentwicklung in der Praxis verfolgt werden könnte.

### Planspielentwicklung als interdisziplinäre Aufgabe

Die Entwicklung von Planspielen bewegt sich im Schnittbereich verschiedener Disziplinen und Fachbereiche.

*Tabelle 1: Planspielentwicklung im interdisziplinären Kontext*

| Disziplin | Fachbereich, Ausschnitt |
|---|---|
| Betriebswirtschaftslehre und verwandte Fakultäten | Versicherungsbetriebslehre |
| | Systemtheorie |
| | Organisationstheorie |
| | Quantitative Betriebswirtschaftlehre, Operations Research |
| Psychologie | Organisationspsychologie |
| | Lernpsychologie |
| Pädagogik | Computer-unterstütztes Lernen |
| | Planspiel als Ausbildungsinstrument |
| | Lerntheorien |
| angewandte Informatik | Objektorientierte Systeme |
| | Simulationsverfahren |
| | Software-Engineering |
| Statistik | Wahrscheinlichkeitsrechnung |
| | Risikotheorie |

19 Es wundert den Leser sicherlich, daß eine neue Software bereits die Versionsnummer 2.0 trägt. Object-VersPlan ist aus den Erfahrungen mit dem konventionell entworfenen und implementierten VersPlan 1.0 entstanden und erhält deshalb die Versionsnummer 2.0. Auf die bei der Arbeit an und mit VersPlan 1.0 gemachten Erfahrungen wird Bezug genommen.

Es ist leicht verständlich, daß bei einer derart breiten Anlage der Fachgebiete nur eine Diskussion der für objektorientierte Planspielentwicklung wichtigsten Aspekte geführt werden kann. Es wird deshalb an entsprechender Stelle statt eingehender Diskussion auf vertiefende Literatur verwiesen.

## Vorgehensweise

1. Das erste Kapitel stellt Motivation und grundlegende Ideen vor. Mit einem Überblick über die wesentlichen Aspekte der objektorientierten Planspielentwicklung soll dem Leser ein erster Einstieg in den Themenkomplex ermöglicht werden.

2. Im anschließenden zweiten Kapitel werden deshalb aus den Disziplinen quantitative Betriebswirtschaftslehre und Pädagogik die für Verständnis und Abgrenzung des Themas notwendigen Grundlagen über Systeme, Modellbildung, Simulation, Planspiele und Planspielentwicklung in aller gebotenen Kürze erörtert.

3. Im dritten Kapitel soll ein Konzept anforderungsorientierter Planspiele erarbeitet werden: Beiträge aus Pädagogik und Psychologie begründen die Motivation für dieses Konzept und legen den methodischen Grundstock der weiteren Vorgehensweise.

4. Das folgende vierte Kapitel beschäftigt sich detailliert mit den Möglichkeiten des objektorientierten Paradigmas zur Entwicklung von Unternehmensplanspielen. Dazu sollen die für das Verständnis der technischen Aspekte notwendigen Grundlagen zu objektorientierten Systemen aus dem Fachbereich der angewandten Informatik dargestellt werden.

5. Im fünften Kapitel werden die theoretischen Überlegungen im Anwendungsfall Versicherungsplanspiel eingebracht. Hier sollen allgemein anwendbare Werkzeuge zur Planspielkonstruktion und -durchführung und ein Application-Framework[20] für Versicherungsplanspiele vorgestellt werden. In diesem Kapitel liegt der Schwerpunkt im Zusammenspiel der betriebswirtschaftlichen Aspekte mit den in den vorangegangenen Kapiteln erarbeiteten Konzepten. So treten die Fachgebiete der Versicherungswirtschaft, der angewandten Informatik und Statistik sowie der Organisationspsychologie immer mehr in den Vordergrund.

6. Das sechste Kapitel schließt mit einer Bewertung und einem Ausblick die Betrachtungen ab.

20 Begriff und Inhalt des Application-Frameworks werden im fünften Kapitel noch näher erläutert.

# 2. Grundlagen zu Planspielen

Im nun folgenden Kapitel werden die für die Beschäftigung mit Planspielen grundlegenden Begriffe erörtert:

1. Mit der Vorstellung des Systemansatzes soll die Grundposition interdisziplinärer Vorgehensweise aufgezeigt werden.

2. Der Abschnitt zu Simulation soll Begriffe, Inhalte und Verfahren von der Modellbildung bis zur Implementierung vorstellen.

3. Die Begriffe Planspiel und Planspielentwicklung sollen im dritten Abschnitt mit Inhalt gefüllt und abgegrenzt werden.

## 2.1. Systeme

System ist ein schillernder Begriff, der in vielen unterschiedlichen Kontexten mit abweichender Bedeutung benutzt wird.

Im Zusammenhang mit der Planspielentwicklung hilft die Annäherung *Butewegs* an dieses Thema weiter: Er versteht unter dem Systemansatz und der Systemforschung die Gesamtheit von Bemühungen, Probleme mit Hilfe systemtheoretischen Gedankengutes lösen zu wollen.[21] Das Grundproblem von Planspielen - die modellhafte Abbildung einer **Mikrowelt**, mit der man in einer Simulationsumgebung spielen und lernen kann - ist geradezu prädestiniert, von Ideen und Erkenntnissen der Systemtheorie zu profitieren. Um dies in späteren Kapiteln diskutieren zu können, sollen im folgenden die Grundgedanken des systemorientierten Ansatzes vorgestellt werden.

### Begriff des Systems

Der Begriff System stammt aus dem Griechischen und meint: Prinzip - Zusammenstellung - Ordnung - Gefüge - Wirkzusammenhang von Dingen und Vorgängen. Diese Aufzählung deutet die konstitutiven Merkmale eines Systems bereits an:[22]

21 Vgl. hierzu und zum folgenden: Buteweg, J. (Systemtheorie, 1988), S. 11

22 Vgl. hierzu und zum folgenden: Meyer, M. (Operations Research, 1990), S. 8 ff.

## Konstitutive Merkmale: Ganzheit, Teile und Beziehungen

Grundlegende Annahme des Systemansatzes ist, daß die Bestandteile von Systemen - Elemente oder Objekte genannt - nicht isoliert zu betrachten sind, sondern vor allem die Wechselwirkungen zwischen ihnen zu sehen sind: Es entsteht aus den Elementen ein Ganzes. Sichtbare Folge dieser Verknüpfungen können beispielsweise Phänomene der Selbstorganisation oder der Selbstregulierung von Systemen sein.
Dies wird treffend in der von *Aristoteles* formulierten Aussage zusammengefaßt, das Ganze sei mehr als die Summe seiner Einzelteile.[23]

## Systemkonzepte

Zur Strukturierung von Systemen haben sich im wesentlichen drei Systemkonzepte herausgebildet:[24]

*Tabelle 2: Strukturkonzepte für Systeme*

| Konzept | Erklärung |
|---|---|
| funktional | Ein System wird als ein schwarzer Kasten (engl. black box) verstanden, der über Eingänge und Ausgänge mit der Umgebung verbunden ist.<br>Über die Beobachtung der Ausgänge nach Manipulation der Eingänge wird auf die Funktion und damit auf die Struktur geschlossen. |
| struktural | Ein System wird als Ganzheit von miteinander verknüpften Elementen gesehen: Die Verknüpfungen symbolisieren die Relationen zwischen den Teilen. |
| hierarchisch | Ein System wird in verschiedene Ebenen zerlegt, die hierarchisch als Supersystem, System und Subsystem verstanden werden.<br>Da Systeme und (nicht mehr weiter verfeinerte) Elemente miteinander verknüpft werden können, steht der hierarchische Ansatz dem strukturalen Ansatz näher als dem funktionalen Ansatz. |

Diese drei Konzepte können zu einem umfassenden Systemkonzept verbunden werden:

Somit kann ein System verstanden werden als ...

1. eine Ganzheit, die Beziehungen zwischen Elementen aufweist, ...

23 zitiert in: Meyer, M. (Operations Research, 1990), S. 8
An dieser Stelle sei auf die Diskussion der Emergenz, den nicht aus den Eigenschaften der Teile erklärbaren Systemphänomene, verwiesen: vgl. Krohn, W., Küppers, G. (Emergenz, 1992), S. 7 ff.

24 Vgl. Ropohl, G. (Systemtheorie der Technik, 1979), S. 49 ff.

2. die ihrerseits aus miteinander verknüpften Teilen oder Subsystemen bestehen können ...

3. und die auf einer bestimmten Abstraktionsebene von ihrer Umgebung abgegrenzt werden können.[25]

Im Abschnitt 4.3 wird genau dieser Aspekt aufgegriffen und für die objektorientierte Planspielentwicklung ausgebaut. Werkzeuge und Techniken, die diese Strukturkonzepte implementieren, werden im fünften Kapitel beschrieben.

### Grundgedanken der allgemeinen Systemtheorie

Vor dem Hintergrund auseinanderdriftender, sich gegenseitig abschließender Einzelwissenschaften und der Überbetonung des analytischen, zergliedernden Denkens entstand die allgemeine Systemtheorie Mitte der vierziger Jahre als integrierend wirkende Gegenbewegung.[26]

Bei der Vorstellung ihrer Grundgedanken soll der Schwerpunkt auf den gemeinsamen Ideen und nicht auf der Abgrenzung der verschiedenen Strömungen und den Systematisierungs- und Formalisierungsbestrebungen liegen.[27]

### Komplexität

Allen Ansätzen gemein ist die Konfrontation mit dem Problem der **Komplexitä**t: Dinge hängen zusammen und beeinflussen einander in einer Art und Weise, die es dem Menschen nicht mehr erlaubt, alle Elemente und alle ihre Verflechtungen zu überblicken und zu begreifen.

Komplexität kann nach *Bronner* mithin als die „Anzahl der Elemente und ihrer Relationen“[28] in einem System verstanden werden. In Abgrenzung zur Komplexität versteht er Kompliziertheit als die „Verschiedenheit der Elemente und Relationen“[29].

---

25 Vgl. Buteweg, J. (Systemtheorie, 1988), S. 21

26 Als Begründer der allgemeinen Systemtheorie gelten *v. Bertalnffy*, *Boulding* und *Rappaport*. Sie widmeten sich vor allem der Formalisierung von Systemen.
Siehe v. Bertalanffy, L. (Systemtheorie, 1962) S. 235 ff.; siehe auch Rapoport, A. (Systemtheorie, 1988)

27 Als die beiden großen, sich strikt voneinander abgrenzenden, Hauptrichtungen der Systemtheorie lassen sich im wesentlichen die funktionalistische Systemtheorie (vertreten u.a. von *Luhmann* und *Parsons*) und die techno-kybernetische Systemtheorie (hervorgegangen aus der allgemeinen Systemtheorie) abgrenzen.
Zur Abgrenzung vgl. Buteweg, J. (Systemtheorie, 1988), S. 11f.

28 Bronner, R. (Komplexität, 1992), Sp. 1122

29 Bronner, R. (Komplexität, 1992), Sp. 1122

Erste Anstöße zur Beherrschung der Komplexität kamen aus der Biologie, die mit eingeschränkten, isolierbaren und wiederholbaren Laborexperimenten versuchte, dem Problem Herr zu werden. Die Wirtschafts- und Sozialwissenschaften haben ebenfalls erkannt, daß Kern vieler ihrer Probleme die Beschäftigung mit der Komplexität ist: Wegen der scheinbar nicht zu beherrschenden Komplexität können für viele Phänomene nur prinzipielle Erklärungen erarbeitet werden.[30]

### Ganzheitliche Sichtweise

Die Anhänger der allgemeinen Systemtheorie formulieren als Antwort auf diese Probleme mit der Komplexität eben nicht die klassische Zerlegung der Problemgebiete in Teile, die im Detail untersucht werden und von denen man dann auf die Gesamtheit schließen kann.[31] Sie postulieren vielmehr ein ganzheitliches Konzept: Das gesamte Untersuchungsobjekt (System) muß im Auge behalten werden. Dessen Eigenschaften und Verhaltensmuster werden nicht nur allein als Folge der Eigenschaften und des Verhaltens seiner Teile, sondern als Folge der Vernetzung seiner Teile angesehen.[32]

### Interdisziplinarität

Unmittelbare Konsequenz der Vernetzung der Systembestandteile ist die gleichzeitige Beschäftigung mit verschiedenen Fachbereichen. Um das Ziel der Formulierung allgemeiner Systemgesetze zu erreichen, sollen nach Ansicht der allgemeinen Systemtheorie in einem ganzheitlichen Ansatz Ähnlichkeiten, Analogien und Isomorphien aus verschiedenen wissenschaftlichen Disziplinen herausgearbeitet werden.

Dieser Ansatz stößt auf Kritik in der Betriebswirtschaftslehre. So formuliert KIRSCH: „Heute ist es in erster Linie Aufgabe eines Ansatzes, die Vielfalt in der Wissenschaft und Praxis anzuerkennen. Nicht die Bändigung der Vielfalt, sondern die Möglichkeiten ihrer Bejahung steht im Vordergrund.“[33]

---

30 Vgl. v. Hajek, F. A. (Theorie komplexer Phänomene, 1972), S. 12 ff.

31 *Schneider* nennt diese Partialmodelle „eine ‘mögliche Welt’ mit kleinem Guckloch davor“. Schneider, D. (Allgemeine Betriebswirtschaftslehre, 1987), S. 58

32 Dies bedeutet aber nicht, daß die Systemtheorie die Bedeutung des Ganzen über die Bedeutung der Teile erhebt. Sie versucht vielmehr, Systemphänomene aus der Vernetzung seiner Teile zu erklären. Vgl. Buteweg, J. (Systemtheorie, 1988), S. 16

33 Kirsch, W. (Entscheidungsorientierte Betriebswirtschaftslehre, 1989) S. 131
*Kirsch* zeigt am Beispiel der Betriebswirtschaftlehre die Tendenz zu heterogenen Forschungsaktivitäten, die zu Teilbereichen anderer Disziplinen relevante Beiträge leisten. Er führt dies auf die Orientierung an praxisrelevanten Problemen zurück, die keine Abgrenzungen zu Disziplinen erlauben. Seine Konsequenz ist die Interpretation der Betriebswirtschaftslehre als eine angewandte, multidisziplinäre Lehre von der Führung. Diese Führungslehre bedarf nach Meinung *Kirschs* zwar einer organisationstheoretischen Fundierung, sie kann sich jedoch in der Beschäftigung mit Managementproblemen nicht auf eine Forschungsrichtung - wie beispielsweise die Mikroökonomie -

Der Systemgedanke und die Überlegungen der allgemeinen Systemtheorie liefern die notwendigen Beiträge, um der Beschäftigung mit komplexen Tatbeständen, wie sie in der Realität vorkommen, gewachsen zu sein. Für die Entwicklungen von Planspielen, die sich die Realität zum Vorbild nehmen, können systemtheoretische Überlegungen deshalb einen wertvollen Beitrag leisten.

Im folgenden Abschnitt wird der Systemgedanke wieder aufgenommen und seine Anwendung in Modellbildung und Simulation diskutiert.

## 2.2. Simulation

Im Zuge neuerer Entwicklungen im Bereich der Informatik und Datenverarbeitung hat sich die Vorstellung von Simulation immer weiter zersplittert: Zunehmend wird Simulation mit neuen Kommunikationsformen wie Cyberspace, Computer Based Training, Hypermedia oder Multimedia in Verbindung gebracht.[34]

Für die im Zusammenhang mit der Entwicklung von Unternehmensplanspielen wichtigen Begriffsdefinitionen und Abgrenzungen macht es Sinn, sich auf die Grundgedanken und Basiskonzepte von Simulation zu konzentrieren.

*Böhret* / *Wordelmann* extrahieren zwei Hauptmerkmale aus den vielfältigen Abgrenzungen, die in der Literatur für den Begriff Simulation zu finden sind: Modellierung und Dynamik[35]- zwei Begriffe, die im folgenden etwas genauer betrachtet werden sollen.

### Modellierung

Intuitiv könnte man Modellierung als Nachahmung und Abbildung der Realität auffassen - in der Literatur wird der Begriff jedoch sehr unterschiedlich verwendet.

Um dem Begriff systematisch zu nähern, kann ein Blick auf die Vorgehensweise der Modellbildung nützlich sein.[36] Unabhängig vom verwendeten Instrumentarium kann Modellierung als Prozeß verstanden werden, der sich in zwei Stufen vollzieht:

---

beschränken.
Vgl. Kirsch, W. (Betriebswirtschaftslehre, 1993), S. 12 ff.

34 Vgl. Liebl, F. (Simulation, 1992), S. 3 ff.

35 Vgl. Böhret, C., Wordelmann, P. (Fortbildung, 1975), S. 19 ff.

36 Vgl. hierzu und zum folgenden Meyer, M. (Operations Research, 1990), S. 16
*Meyer* baut auf den Überlegungen *Kosiols* auf.

1. Zuerst wird ein abgegrenzter und überschaubarer Teilzusammenhang charakteristischer Tatbestände zweckorieniert, abstrahierend und vereinfachend ausgegliedert: Es wird das zu modellierende System abgesteckt.[37]

2. Dann wird dieses Gedankengebäude durch Abbildung mit Hilfe eines geeigneten Mediums in einer der Betrachtung zugänglichen und intersubjektiv überprüfbaren Form dargestellt und mitgeteilt. Es entsteht ein Systemmodell.[38]

Im Zentrum der Modellbildung steht der Abbildungsaspekt: Die Transformation vom Wirklichkeitsausschnitt in das Modell. Im Vordergrund stehen dabei in aller Regel der Aspekt der Vereinfachung und die Forderung nach Strukturgleichheit oder -ähnlichkeit zwischen Realsystem und Modell. Vereinfachung bedeutet eine mehreindeutige Abbildung der Elemente des Gegenstandsbereichs in das Modell - und genau hier kommt der Anwendungszweck zum Tragen: Es werden nur die jeweils relevanten Aspekte des Realsystems erfaßt. Eine weitere Vereinfachung kann durch die Verwendung von Äquivalenzklassen statt vieler gleichartiger Elemente im Modell erreicht werden. Die Forderung nach Strukturgleichheit oder -ähnlichkeit kann durch eine eineindeutige Abbildung der Systemverbindungen im Modell realisiert werden.[39]

Der weitergehende Anspruch nach Isomorphie[40] zwischen Modell und Realität erfährt in der Literatur breite Ablehnung, da das Ziel der Vereinfachung nicht erreicht werden kann. [41]

Das Abgrenzen und Ausgliedern der relevanten Teilzusammenhänge und die Abbildung als Systemmodell sollen im Abschnitt 2.3 am Beispiel der Planspielentwicklung diskutiert werden. Dabei muß man bedenken, daß Systemabgrenzung und Modellbildung immer vor dem Hintergrund der Perspektive des mit dem Problem Beschäftigten erfolgt.[42]

---

37 *Schneider* streicht den beschränkenden Aspekt dieser Auschnittsbildung heraus: „Jedes Modell ist eine unvollständig beschriebene mögliche Welt, in welcher der Rest der Welt, nach dem sich die empirische Wahrheit oder Falschheit einer jeden Aussage richten, überhaupt nicht auftritt.“ Schneider, D. (Allgemeine Betriebswirtschaftslehre, 1987), S. 58

38 Vgl. Meyer, M. (Operations Research, 1990), S. 16 f.
Inwieweit das objektorientierte Instrumentarium dieses geeignete Medium sein kann, soll im vierten Kapitel aufgegriffen werden.

39 Vgl. Bamberg, G., Coenenberg, A. (Entscheidungslehre, 1981), S. 13 f.

40 Isomorph kommt aus dem Griechischen und bedeutet soviel wie ‘von gleicher Gestalt’.

41 Vgl. Bamberg, G., Coenenberg, A. (Entscheidungslehre, 1981), S. 14

42 Eine bedeutsame Konsequenz aus dieser Erkenntnis ist zum Beispiel, daß die Wiederverwendung von Elementen durch unterschiedliche Auffassungen und Lösungsansätze unterschiedlicher Ersteller erschwert wird - ein Aspekt, der am Beispiel der Planspielkonstruktion noch einmal aufgegriffen wird.

## Aussagekraft und Genauigkeit eines Systemmodells

Die allgemeine Systemtheorie hat mit dem Vorurteil zu kämpfen, ihr Ansatz sei so allgemein ('Alles kann als System betrachtet werden'), daß dieser empirisch leer würde. Die Systemtheorie ist keine empirische Theorie: „Sie ist vielmehr ein Kalkül (...), und zwar ein teilinterpretierter Kalkül, denn neben dem rein abstrakt mathematischen Relationengebilde wurden einige Grundbegriffe eingeführt, wie Umwelt, Input, Output, die gewisse Vorstellungen über empirische Systeme bedingen.“[43]

*Vester* greift die Diskussion nach dem notwendigen Grad der Vereinfachung bei der Anwendung dieses Kalküls für ausgewählte Probleme auf: Er verweist auf die Dimension und Komplexität realer Systeme und stellt die Frage, in wieweit angesichts der Fülle der zu modellierenden Systembestandteile, Variablen und Zusammenhänge die Komplexität des Modells der Realität angeglichen werden muß. Mit Hilfe eines Systemmodells ausgesuchter Schlüsselvariablen[44] bekommt man nach Ansicht *Vesters* ein wesentlich getreueres Abbild eines realen Systems, als man es durch die, auch noch so detaillierte, Betrachtung der Entwicklung einer oder weniger Hauptvariablen (etwa dem Bruttosozialprodukt) bekommen würde. Mit Systemmodellen können auch Prozesse betrachtet werden, die nicht gleichmäßig und stetig verlaufen.[45]

Er bringt die Zweckorientierung von Modellen ins Spiel: Systemmodelle sollen durch deren Möglichkeiten für die Beschreibung und Erklärung von Systemen Verständnis für das Funktionieren von realen Systemen erzeugen. Entscheidungen und Folgen auf das System können so besser abgeschätzt und optimiert werden. Systemverhalten läßt sich, so *Vester*, nicht durch eine isolierte Betrachtung von Einzeleingriffen, sondern nur durch die Erfassung der Gesamtkonstellation erkennen. Dieser ganzheitliche Erkennungsprozeß entspricht dem Verständnis der Mustererkennung des menschlichen Gehirns[46].

Folgt man der Einschätzung *Beers*, so sind die äußerst komplexen, propabilistischen Systeme, wie Unternehmen, Teilmärkte oder gar die ganze Volkswirtschaft, nicht mehr vollständig beschreibbar.[47] Dies ist jedoch kein Grund, sich nicht mit solchen Systemen

---

43 Buteweg, J. (Systemtheorie, 1988), S. 41

44 Vgl. Vester, F. (Neuland, 1991), S. 46
*Vester* spricht noch von Schlüsselvariablen, weil auch er in der Tradition klassischer System Dynamics Modelle (siehe Abschnitt 2.2) argumentiert. In objektorientierter Denkweise würde man von Schlüsseleigenschaften und -verhalten von Systemen und Systembestandteilen (beides sind Objekte) sprechen.

45 Vgl. hierzu und zum folgenden: Vester, F. (Neuland, 1991), S. 46 ff.

46 Vgl. Vester, F. (Neuland, 1991), S. 37
Gerade dieser Erkennungsprozeß macht aber den allermeisten Menschen Probleme: *Dörner* zeigt anhand zahlreicher Fallstudien die Probleme mit dem Entscheidungsverhalten in komplexen, vernetzten, dynamischen Systemen.
Siehe Dörner, D. (Strategie, 1993)

47 Siehe Beer, S. (Kybernetik und Management, 1963) zitiert von Meyer, M. (Operations Research, 1990), S. 11f.

zu beschäftigen - es müßten vielmehr Methoden geschaffen und Fähigkeiten geschult werden, auch solche Systeme handhabbar abzubilden und so einen Erkenntnisfortschritt aus der Beschäftigung mit ihnen ziehen zu können.

*Vester* vertritt deshalb die Ansicht, daß in einem Systemmodell zwanzig oder dreißig ausgewählte Schlüsselvariablen ausreichen, wenn nur die grundlegenden Zusammenhänge in diesem Modell richtig erkannt und beschrieben wurden.[48] Eine Lösung wäre, von den oben angesprochenen Strukturkonzepten Gebrauch zu machen: Man beschreibt die Vernetzung auf Abstraktionsebenen, deren Verhalten und Struktur bekannt sind und beschränkt sich bei ausgewählten Elementen und Subsystemen auf globale Modellierung als black box. Dies wird dann in Betracht kommen, wenn Elemente oder Systeme nicht im Detail bekannt oder für den Einsatzzweck des Modells (zum Beispiel als Lerninstrument) nicht von Belang sind.

### Dynamik

Der zweite wesentliche Aspekt für Simulationen ist die Dynamik: *Rapoport*, ein wichtiger Vertreter der allgemeinen Systemtheorie, nennt Dynamik „das Studium von Systemen, deren Zustände sich ständig ändern“[49]. Je nachdem, wie man den Begriff der Zustände auslegt, kann man Dynamik in einander aufbauenden Stufen zunehmender Komplexität betrachten:

1. Einfachster Fall der Dynamik ist die qualitative Dynamik:[50] Hier steht die Betrachtung der kausalen Zusammenhänge im Vordergrund. Eine qualitativ dynamische Simulation kann beispielsweise Fragen beantworten helfen wie „In welcher Form wirkt sich ein verstärktes Engangement am Markt auf die Kapitaldecke aus?“ oder „Zahlen sich verstärkte Anstrengungen für die Bestandspflege durch geringere Schadenquoten aus?“.

2. Bringt man den Faktor Zeit mit ins Spiel, so kann von quantitativer Dynamik gesprochen werden:

   – Einfachste Form der Betrachtung zeitlicher Aspekte ist die in den Wirtschaftswissenschaften verbreitete komparativ-statische Vorgehensweise: Zeitpunktbetrachtungen der Variablen und des Verhaltens der Systemelemente werden in Zeitreihen erfaßt. Diese Art der Einbeziehung der Zeit spielt vor allem für hochaggregierte Betrachtungen eine Rolle, die sich - wie beispielsweise die Makroökonomie oder die Katastrophentheorie - vorrangig mit Gleichgewichtszuständen beschäftigen. Mit komparativ statischer Betrachtungsweise können nur Systeme mit langsamer Makrodynamik und schneller Mikrodynamik untersucht

48 Vgl. Vester, F. (Neuland, 1991), S. 45 f.

49 Rapoport, A. (Systemtheorie, 1988), S. 38

50 Zum Dynamikbegriff: vgl. hierzu und zum folgenden Hanisch, H.-M. (Petri-Netze, 1992), S. 65 ff.

werden. Zeigt ein System hingegen schnelle Makrodynamik - ändern sich also Strukturen auf hohem Aggregationsniveau sehr schnell - so kann keine Aussage mehr gemacht werden.[51]

- Komplexer wird die Modellierung, wenn die exakten Zeitpunkte von Ereignissen und die Dauer von Aktivitäten betrachtet werden: Diese Art der Modellierung ermöglicht auch die Abbildung schneller Dynamik und steht dem mikroökonomischen Ansatz näher.[52]

- Wenn darüberhinaus noch Strukturveränderungen dynamisch erfaßt werden, können in solchen Systemen Entwicklungen der Organisation bis hin zu Phänomenen der Selbstorganisation abgebildet werden.[53]

## Simulation

*Böhret* / *Wordelmann* definieren aus diesen Abgrenzungen heraus Simulation allgemein als „ ... Methode der Beobachtung und Analyse von Entwicklungen variierender Inputs (Systemzustände) innerhalb dynamischer Modelle („Simulationsmodelle“) ...“[54]. In dieser Definition spiegelt sich ihre Sicht von Systemen wider: Sie gliedern das zugrundeliegende Gesamtmodell in einen Aktionsbereich, in dem menschliche Einflußnahmen und Aktivitäten stattfinden können und einen Reaktionsbereich, der den formalisierten Teil des Modells darstellt.

Simulation im obigen Sinne wird in den im folgenden vorgestellten Simulationsverfahren umgesetzt und konkretisiert.

51 Vgl. Rapoport, A. (Systemtheorie, 1988), S. 66 ff.

52 Eine detaillierte Diskussion der Mikro- / Makroproblemtik findet sich im vierten Kapitel.

53 Zur Selbstorganisaiton vgl. Krohn, W., Küppers, G. (Emergenz, 1992), S. 10 ff.

54 Krohn, W., Küppers, G. (Emergenz, 1992), S. 22

## Simulationsverfahren

Der Abgrenzung *Liebls* folgend, lassen sich Simulationsverfahren nach vier Merkmalen unterscheiden: [55]

*Tabelle 3: Abgrenzung von Simulationsverfahren*

| Abgrenzung | | |
|---|---|---|
| nach der Identität der Zeit von Modell und Realität | Echtzeit-Simulation | Zeitraffer-Simulation |
| nach dem zeitlichen Verhalten der Zustandsvariablen im Modell | diskret | kontinuierlich |
| nach der Vorbestimmtheit des Modells | deterministisch | stochastisch |
| nach der Einbeziehung der Zeit in das Modell | statisch | dynamisch |

## Simulationen in Echtzeit oder im Zeitraffer?

Echtzeit-Simulationen, die reale Vorgänge zeitlich identisch und interaktiv abbilden, wurden zu Anfang hauptsächlich im militärischen Bereich für die Ausbildung von Piloten in Flugsimulatoren entwickelt. Mit der geringer werdenden Bedeutung der Militärs seit Lockerung der Ost-West-Blockbildung und der sehr schnell fortschreitenden Technologieentwicklung der preiswerten, grafikfähigen Computer haben diese Simulationen einen Eingang in die Welt der Spiel- und Homecomputer gefunden. Für Anwendungen im Bereich der Wirtschaftswissenschaften sind Echtzeit-Simulationen wegen der langen Beobachtungszeiträume und der größeren Globalität der Modellierung jedoch nur in ausnahmefällen angezeigt. Werden in diesem Bereich Simulationen benötigt, so sollte deshalb der zeitliche Ablauf im Zeitraffer wiedergegeben werden. Damit wird es möglich, auch mehrere Szenarien von in der Realität sehr langfristig ablaufenden Geschehnissen zu modellieren.

## Kontinuierliche oder diskrete Systeme

In kontinuierlichen Systemen verändern sich die Zustandsvariablen über die Zeit hinweg stetig: Solche Systeme werden vor allem dann eingesetzt, wenn komplexe Rückkoppelungseffekte (z.B. in hochaggregierten volkswirtschaftlichen Simulationen oder zur Beschreibung von Problemen aus den Naturwissenschaften) abgebildet werden sollen. Mathematische Basis dieser Modelle sind meistens Differen-

55 Vgl. hierzu und zum folgenden Liebl, F. (Simulation, 1992), S. 3 ff.

zen- und Diffentialgleichungen, wie sie etwa dem ‘System Dynamics-Ansatz’ von *Forrester*[56] zu Grunde liegen.

Für niedrig aggregierte, sozio-technische Systeme, wie sie auch in Unternehmensplanspielen modelliert werden, eignet sich eher eine diskrete Beschreibung der Zustandsvariablen: Die Beobachtungen der Ausprägungen der Zustandsvariablen in der Realität sind diskret darstellbar. In vielen Fällen macht eine stetige Repräsentation gar keinen Sinn: Eine Maschine ist zu einem Zeitpunkt belegt oder nicht, Schadenzahlen haben ganzzahlige Ausprägungen.

### Deterministische oder stochastische Simulation?

In deterministischen Simulationen werden Zustandsvariablen nach bestimmten Regeln von Periode zu Periode fortgeschrieben. Selbst deterministische Simulationsmodelle, die aus einer relativ kleinen Zahl von Basisoperationen bestehen (so etwa das bekannte ‘game of life’), können durch die Einbeziehung der Zeit eine Vielfalt von Verhaltensmustern aufweisen. Beispiele für deterministische Simulationen, die dadurch hohe **Komplexität** erreichen, sind Markt- oder Bilanzsimulationen, die in der Regel Bestandteil eines Unternehmensplanspiels sein werden.[57]

Stochastische Simulationen tragen dem Umstand Rechnung, daß Zufallsphänomene der Realität Einzug in die modellierte Welt der Simulation halten sollen: Zustandsvariablen werden in stochastischen Simulationen nicht als feste Größe, sondern als Zufallsvariable angegeben. Basis hierfür ist in der Regel die algorithmische Erzeugung von Zufallszahlen.[58]

### Statische oder dynamische Simulation?

Die meisten Simulationen bilden ein Modell im zeitlichen Verlauf ab. Man spricht deshalb von dynamischen Simulationen. Eine Ausnahme ist die statische Monte-Carlo-Simulation, bei der mit Hilfe von Zufallszahlen deterministische, statische Probleme gelöst werden.[59] Im Grenzbereich zwischen Monte-Carlo-Simulation und diskreter, dynamischer Simulation sind die quantitative Risikoanalyse (Risikoanalyse von *Hertz*[60]) und die Cross-Impact-Analyse[61] einzuordnen. Die Betriebs-

---

56 Siehe Forrester, J. W. (Dynamics, 1968); siehe auch ders. (Regelkreis, 1971)

57 LIEBL stellt mit Marktsimulation und Bilanzsimulation ebenfalls diese beiden Beispiele für deterministische Simulationen vor: vgl. Liebl, F. (Simulation, 1992), S. 15 ff.

58 Für die algorithmische Erzeugung von Zufallszahlen: siehe Schmitz, N., Lehmann, F. (Monte-Carlo, 1976)

59 Zwei Anwendungen der Monte-Carlo-Methode sind ‘Hit-or-Miss-Monte Carlo’ und ‘Crude-Monte Carlo’: vgl. Liebl, F. (Simulation, 1992), S. 55 ff.

60 Siehe Hertz, D. B., Thomas, H. (Risk Analysis, 1984)

61 Vgl. Liebl, F. (Simulation, 1992), S. 78 ff

wirtschaftslehre fordert nach Zeiten statischer und statisch komparativer Betrachtungen zunehmend dynamische Modelle und Simulationen.[62]

### Diskrete Simulation

Wie bereits angesprochen, bietet sich für Simulationsmodelle, die sozio-technische Modelle mit mittlerem bis niedrigem Abstraktionsgrad abbilden, der Einsatz der Methoden diskreter Simulation an. Die Tabelle zeigt eine Übersicht der Methoden:[63]

*Tabelle 4: Diskrete Simulation*

| diskrete Simulation | | |
|---|---|---|
| | zeitgesteuert | quasikontinuierlich |
| | ereignisgesteuert | ereignisorientiert |
| | | aktivitätsorientiert |
| | | prozeßorientiert |
| | | transaktionsorientiert |

Diskrete Simulationsmodelle bilden physische Modelle der Realität ab, die sich in der Zeit entwickeln, aus interagierenden Objekten und Prozessen bestehen und durch Zustände, Aktivitäten und Zustandsänderungen beschrieben werden.

### Zeitgesteuert oder ereignisgesteuert?

In zeitgesteuerten Simulationsmodellen wird die Zeit in festen Schritten fortgeschrieben und alle Zustandsänderungen werden global errechnet. Extremfall einer zeitgesteuerten Simulation ist die quasikontinuierliche Simulation.

Unter einem Ereignis im diskreten Modell verstehen *Frauenstein* / *Pape* / *Wagner* eine Zustandsänderung eines Systembestandteils zu einem bestimmten Zeitpunkt. Ereignisse selbst verbrauchen keine Zeit. Je nachdem, welcher Perspektive bei der Modellierung die größte Bedeutung zukommt, unterscheiden sie vier Typen ereignisgesteuerter, diskreter Simulation:

– In der ereignisorientierten Simulation liegt der Schwerpunkt auf einer zentralen Ereignisliste, die Ereignisse (nach ihrem Eintrittszeitpunkt geordnet) enthält und deren Abarbeitung koordiniert.

62 Vgl. Helten, E. (Prognosemethoden, 1976), S. 440

63 Vgl. hierzu und zum folgenden: Frauenstein T., Pape U., Wagner, O. (Sprachkonzepte, 1990), S. 21ff.

– Als weitere Verfeinerung des Ereignisbegriffs führen die Autoren die Begriffe Aktivität, Prozeß und Transaktion ein. Prozeßorientierte Modelle bilden zeitlich geordnete Folgen von Aktivitäten, die meist einem bestimmten Objekt zugeordnet wird. Die prozeßorientierte Methode[64] ermöglicht eine problemgerechte Strukturierung des Modells und erweist sich mächtiger als die Modellierung von Aktivitäten und Transaktionen.

Wie diese Ansätze diskreter Simulation in der Planspielentwicklung ihren Niederschlag finden können, wird im Abschnitt 4.2 noch genauer untersucht.

Zusammenfassend: Ein Simulationsmodell kann als „symbolisches Abbild der Realität“[65] verstanden werden. Es bildet nicht nur die Struktur der Wirklichkeit ab, sondern ist in aller Regel auch ein dynamisches Modell, dessen Eigenschaften und Struktur sich im Zeitablauf verändern.

Für die Abbildung von Systemmodellen mit betriebswirtschaftlichem Kontext eignen sich computerunterstützte, diskrete, ereignisgesteuerte, dynamische Simulationmodelle.

64 Durch Prozesse können logisch zusammengehörige Aktivitäten zu Simulationsobjekten und zu Ereignissen zugeordnet werden. Damit erreicht man einerseits die Lokalisierung von Aktivitäten und Daten und andererseits noch eine zusätzliche Abstraktion über die Methode der Ereignissteuerung.

65 Vester, F. (Neuland, 1991), S. 105

„Das Planspiel oder Im Sandkasten sieht man klarer“[66]

## 2.3. Planspiele und Planspielentwicklung

So titelt der Bericht von einem Rückversicherungsplanspiel der Bayerischen Rückversicherung. Doch welche Merkmale kennzeichnen das Instrument Planspiel, das es hier möglich gemacht hat, sechzehn professionelle Erstversicherer zum gemeinsamen Spiel zusammen zu bringen?

Man kann sich Begriff und Inhalt von Planspielen von zwei Seiten nähern: Konstitutiv durch die Identifizierung von Merkmalen und situationsbezogen durch die Betrachtung des Planspiels und in einer Spielumgebung.

### Konstitutive Merkmale von Planspielen

In Anlehnung an *Rohn* sind es drei Merkmale, die Planspiele charakterisieren: Kontext, Repräsentation und Eingriffsmöglichkeiten.[67]

- Kontext: Basis eines Planspiels ist eine Sammlung von Fakten und Daten eines Ausschnittes aus Wirtschaft, öffentlicher Verwaltung und der Gesellschaft - den *Rohn* als „Kontext“[68]bezeichnet.

- Repräsentation: Wie diese Faktensammlung repräsentiert wird, bestimmt die Anwendung - der Planspiel-Typ: Fakten und Daten können als lose Aufbereitung von Informationen, geordneter Komplex von Daten und Vorereignissen oder dem Systemansatz folgend als Systemmodell angeordnet werden.

66 Bayerische Rück (Das Planspiel, 1982), S. 1

67 Vgl. Rohn, W. (Methodik, 1980), S. 9

68 Rohn, W. (Methodik, 1980), S. 9

– Die folgende Grafik zeigt die Abhängigkeit von Kontext und Repräsentation:

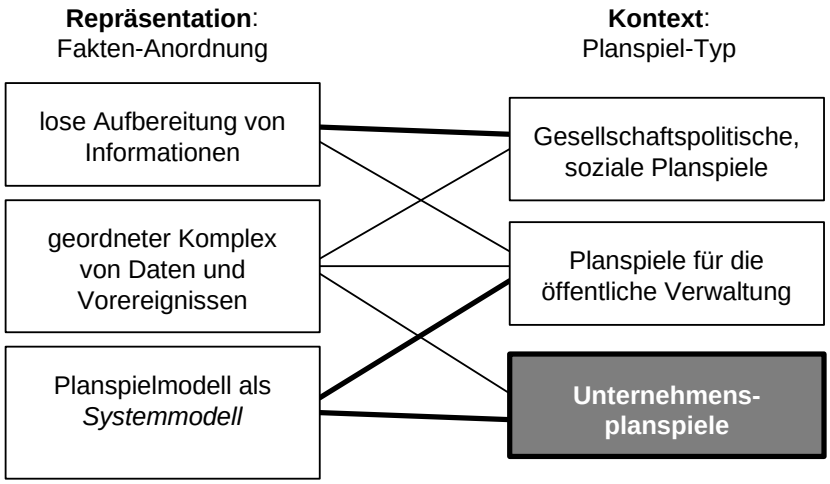



*Abbildung 1: Repräsentation des Kontextes in Planspielen*

Unternehmensplanspiele sind die klassische und häufigste Anwendungsform von Planspielen. Im Zuge der verbesserten Möglichkeiten quantitativer Betriebswirtschaftslehre durch mathematische Verfahren und Computerunterstützung konnten für Planspiele anwendbare Unternehmensmodelle abgebildet werden.[69] In der vorliegenden Arbeit werden diese auf der Systemtheorie basierenden Beiträge aufgegriffen und mit den Möglichkeiten der objektorientierten Methodik weiterentwickelt.

– Eingriffsmöglichkeiten: Alle Arten von Planspielen sind offene Systeme, in die von außen eingegriffen wird: Von den Teilnehmern in Form von Entscheidungen und von der Spielleitung durch Variation der Modellierung (Parameter, Systemeigenschaften).

Doch ein Planspiel kann weit mehr sein, als nur eine in einem Modell anwendungsgerecht repräsentierte Mikrowelt, die Eingriffsmöglichkeiten von außen bietet.

### Spielsituation

Bei Planspielen wirken personale Komponenten und inhaltlich-technische Aspekte zusammen: In einer Spielsituation agieren Spieler und Spielleiter in dem vom Modell

69 Vgl. Rohn, W. (Methodik, 1980), S. 9

abgesteckten Rahmen. *Böhret* / *Wordelmann* sprechen deshalb vom Planspiel als eine Form von „Mensch-Maschine-Simulation“[70].

Die Abbildung zeigt die Elemente der Spielsituation eines Unternehmensplanspiels:

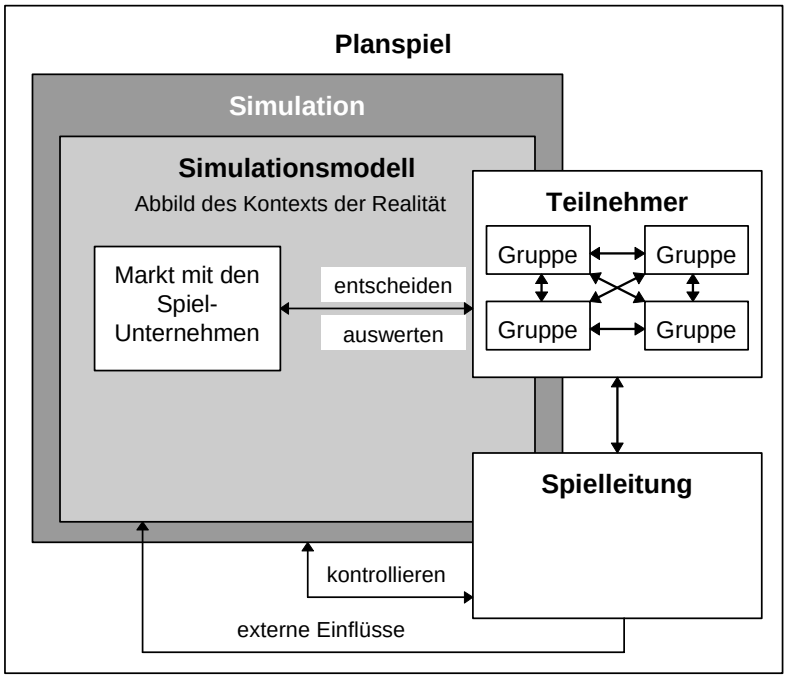



*Abbildung 2: Spielsituation*

## Teilnehmer und Gruppen

Viele Planspiele sind dafür ausgelegt, daß sich die Teilnehmer in Gruppen organisieren: Die Vorgänge innerhalb und zwischen den Gruppen bilden ein maßgebliches Element der Spielsituation.[71] Für die Konstruktion des Planspielmodells spielen diese Verhaltensaspekte mittelbar eine Rolle: Die Möglichkeiten der Modellbildung müssen den nötigen Spielraum für die Spielleitung zur Schaffung der passenden Rahmenbedingungen bieten. Auf diesen Aspekt wird im dritten Kapitel noch genauer eingegangen.

---

70 Böhret, C., Wordelmann, P. (Fortbildung, 1975) S. 27
Es soll hierbei nur der Begriff der ‘Mensch-Maschine-Simulation’ aufgegriffen werden. Wie im folgenden noch näher ausgeführt wird, sollen bei objektorientierter Planspielentwicklung die Spielleitung und die Spieler mit in das System Planspiel eingebunden werden.

71 *Schmidt* und *Helten* / *Schmidt* zeigen am Beispiel des Unternehmensplanspiels Versicherungen die Möglichkeiten auf, mit dem das Instrument des Planspiels neben den fachlich-kognitiven Aspekten auch soziale Kompetenz zu vermitteln.
Vgl. Schmidt, H. (Schlüsselqualifikationen, 1994), S. 1 ff.; vgl. a. Helten, E., Schmidt, H. („Spiel“ Versicherung), S. 85 ff.

### Werkzeuge

Zur Konstruktion und Kalibrierung des Kontextmodells, zur Realisierung der Eingriffsmöglichkeiten in das Modell und zur Durchführung der Simulation in der Spielsituation sind Werkzeuge notwendig. Der Abschnitt 5.2 widmet sich diesem Aspekt detailliert.

### Unternehmensplanspiele im systemorientierten Verständnis

Wie oben bereits angesprochen wurde, wird der Kontext eines betriebswirtschaftlichen Planspiels in der Regel in einem Systemmodell repräsentiert. Doch folgt man dem Systemkonzept (vgl. 2.1), muß diese Vorstellung erweitert werden:

Ein Planspiel kann in seiner Ganzheit als Systemmodell gesehen werden.[72]

Wie auch Abbildung 2 zeigt, finden sich alle in der Spielsituation vorkommenden Elemente im Modell wieder:

- Das Systemmodell Planspiel besteht aus den Teilsystemen Simulationsmodell (der Abbildung des dem **Lernziel** entsprechenden Realitätsausschnittes), Teilnehmer und Spielleitung. Dieses als Kontextmodell bezeichnete Teilsystem ist Bestandteil der Simulation.

- Teilnehmer schlüpfen in die Rolle von Führungskräften der Unternehmen, die sich ihrerseits im Simulationsmodell wiederfinden. Die Unternehmen können ihrerseits wieder als Subsystem oder als black-box implementierte Elemente sein. Der Zweck der Teilsysteme Unternehmung definiert sich aus den strategischen Plänen der Teilnehmer und den Vorgaben des Seminarkonzepts.

- Die Spielleitung kann aus Sicht des Simulationsmodells als Externalität gesehen werden. Spielleiter kontrollieren die Simulation und liefern Input für das System des Kontextmodells: Sie treffen beispielsweise Entscheidungen zur Wirtschaftsentwicklung oder zu der allgemeinen Risikosituation.

### Planspielentwicklung

Nachdem die vorangegangenen Erläuterungen den Begriff des Planspiels definiert und abgegrenzt haben sollten, stellt sich die Frage nach der Planspielentwicklung:

---

72 Die Vorstellung des Planspiels als System kann in Analogie zu *Kirschs* Vorstellung, daß Wirtschaft und Wirtschaftseinheiten als System zu begreifen sind, gesehen werden: Im Zentrum stehen Systeme, Systemelemente und ihre Verknüpfungen und - für die Managementlehre von zentraler Bedeutung - die Entscheidungsprozesse, die auf das Geschehen in diesen Systemen Einfluß nehmen.
Vgl. Kirsch, W. (Betriebswirtschaftslehre, 1993), S. 39 ff.

Welche Probleme müssen auf dem Weg zu einem Planspiel bewältigt werden?

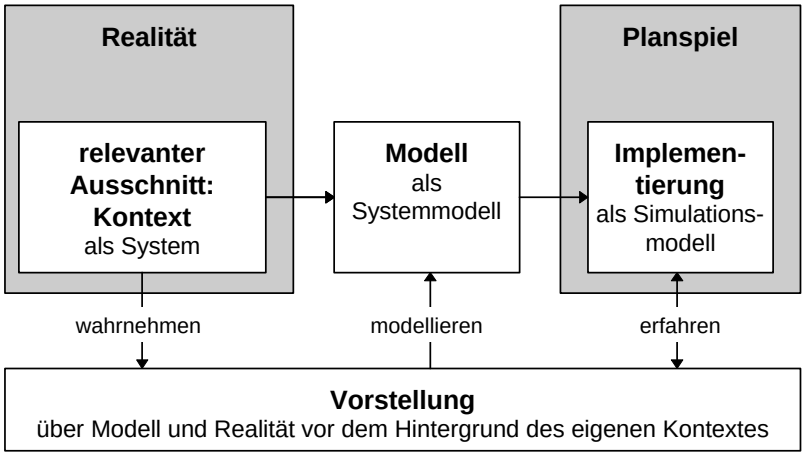



*Abbildung 3: Die Problemfelder der Planspielentwicklung*

### Planspielentwicklung als Abbildungsproblem

Bei der Planspielentwicklung sieht man sich im wesentlichen mit zwei Abbildungsproblemen konfrontiert: Modellbildung und Implementierung.

Die Modellbildung, also die Abbildung eines relevanten Ausschnittes aus der Realität in ein Systemmodell, legt die inhaltliche Basis für ein Planspiel. Es gilt, die für den Einsatzzweck passenden Abstraktionen und Vereinfachungen zu finden und die Vernetzungen der Elemente realitätsnah abzubilden (vgl. 2.2).

Implementierung ist die Abbildung dieses Systemmodells in ein Simulationsmodell und die Erstellung der für Planspielentwicklung und -durchführung notwendigen Werkzeuge. Im wesentlichen wird dies die Abbildung auf die Möglichkeiten der Methoden und Techniken der Softwareentwicklung sein. Im vorliegenden Beispiel wird die objektorientierte Methodik im vierten und fünften Kapitel zeigen, inwieweit sie diesem Abbildungsproblem gewachsen ist.

### Planspielentwicklung als Erkenntnisproblem

Planspielentwicklung kann in mehrfacher Hinsicht als Erkenntnisproblem verstanden werden:

– Welcher Ausschnitt aus der Realität - der Kontext[73] - wird für die mit dem Planspiel zu erfüllende Aufgabe benötigt?

73 *Krohn* / *Küppers* sprechen vom „Forschungskontext“, der durch ein spezielles „Erkenntnisinteresse“ aus dem Realitätsbereich ausgegrenzt wird.
Krohn, W., Küppers, G. (Emergenz, 1992), S. 12

Dieses Problem mündet in die Diskussion über ein Konzept anforderungsorientierter Planspielentwicklung, wie es im dritten Kapitel erarbeitet wird.

– Welche Elemente, mit welchen Eigenschaften, welchem Verhalten und welchen Beziehungen zu anderen Elementen und Systemen konstituieren dieses System ? Die Methoden der objektorientierten Analyse, wie sie im vierten Kapitel vorgestellt werden, widmen sich diesem Problem.

Systemanalyse, Ausschnittsbildung und Modellierung finden als Erkenntnisprozeß vor dem Hintergrund des Kontextes der damit beschäftigten Personen statt. Dem Stufenmodell der Modellbildung von *Stachowiak* folgend, vollzieht sich Modellierung in mehrstufigen Filterprozessen, in Ketten von Abstraktionen (vom Speziellen zum Allgemeinen) und Differenzierungen (vom Allgemeinen auf Spezielles schließen). Dieser Prozeß läuft solange, bis (a) bekannte Begriffe und Sachverhalte erkannt werden, (b) Analogien zu bekannten Sachverhalten identifiziert werden können oder (c) starke, und damit den Sachverhalt charakterisierende, Unterschiede zu bekannten Begriffen gefunden werden.[74]
Im Hinblick auf die Zielorientierung von Planspiel-Simulationsmodellen sind diese Anmerkungen in mehrfacher Hinsicht bemerkenswert:[75]

1. Das Simulationsmodell ist ein Beschreibungsmodell des Realitätsausschnittes. Es dokumentiert Eigenschaften, Verhalten und Vernetzungen von Systemen und ihren Elementen. Es bildet den Hintergrund für das Handeln der Teilnehmer in der Spielsituation. Dieser Charakter des Modelles kommt besonders dann zum Tragen, wenn die Teilnehmer über Eigenschaften, Verhalten und Vernetzungen der Realität lernen wollen: Beispiele für Fragen, die ein Beschreibungsmodell in einem Versicherungsplanspiel beantworten könnte, sind etwa ‘Wie kann ein Versicherungsunternehmen organisiert sein?’ oder ‘Welche Marktteilnehmer konkurrieren mit welchen Produkten und mit welchen Marketingentscheidungen auf dem Markt ... ?’.
   Erfahrung und Sachkenntnis des Modellierenden legen die obere Schranke für die Vermittlung von Systemeigenschaften und -verhalten direkt aus dem Modell. Doch ein Planspiel bietet über die Gruppensituation und die Einbindung von Teilnehmern mit unterschiedlichen Erfahrungen und Kenntnissen die Möglichkeit über die Bildung von Analogien zu weitergehenden Erklärungen. So könnten Teilnehmer untereinander Erfahrungen austauschen: ‘In der Realität funktioniert das im Prinzip genauso wie hier im Modell, aber die Aspekte ... und ... müßten noch zusätzlich berücksichtigt werden’.[76]

74 Stachowiak, H., 1973, zitiert und bearbeitet von: Freiburghaus, M. (Simulation betriebswirtschaftlicher Systeme, 1993), S. 8 f.

75 Zur Unterscheidung in Beschreibungsmodelle und Erklärungsmodelle:
Vgl. Bamberg, G., Coenenberg, A. (Entscheidungslehre, 1981), S. 13 f.

76 Lernumgebungen und Verhaltensaspekte werden im Abschnitt 3.1 aufgegriffen.

Bei der Konstruktion eines Planspiel-Simulationsmodells können über Arbeitsteilung und Wiederverwendung von Modellen, Teilmodellen und Modellbestandteilen, die von unterschiedlichen Erstellern erarbeitet wurden, die Erklärungsmöglichkeiten erheblich ausgeweitet werden. Mehrere Ersteller bedeuten aber auch mehrere Perspektiven - zu der Chance der Einbeziehung zusätzlicher Erfahrungen kommt das Problem der Kompatibilität der Sichten. Ein Aspekt, der über den objektorientierten Ansatz zwar formalisiert werden kann, aber zusätzlich zum Beispiel durch Prozesse der Quasi-Standardisierung der Vorstellung von bestimmten Ausschnitten der Wirklichkeit - also auch von Fachbereichen („herrschende Meinung“) - entschäft werden muß.

2. Darüberhinaus soll dieses Modell den Teilnehmern auch erklären, warum Eingriffe und Veränderungen nun gerade diese Konsequenzen hervorgebracht haben: Man spricht deshalb auch von Erklärungsmodellen. Für ein Planspiel bedeutet dies, daß die Teilnehmer die Auswirkungen ihrer und anderer Entscheidungen nachvollziehen können: Dazu gehört einerseits eine realitätsnahe Implementierung der Vernetzung der Systemelemente und andererseits die Möglichkeit, die Entwicklungen und Wirkzusammenhänge auch zu kommunizieren.[77]
   Auch für die Anwendung als Erklärungsmodell sind die oben angesprochenen Aspekte der Wiederverwendung bedeutsam. Zusätzlich kommt aber hinzu, daß durch die Beschäftigung der Ersteller mit dem Modell durchaus Eigenschaften und Verhalten entdeckt werden können, die a priori nicht bekannt waren. Dies streicht die Bedeutung der Dokumentation noch einmal heraus.

### Planspielentwicklung als Prozeß

Planspielentwicklung kann als Prozeß von den Anforderungen bis hin zum fertigen Planspiel betrachtet werden. Hierbei wird auch die Problematik der Einbeziehung subjektiver Aspekte wieder aufgegriffen.

Im Abschnitt 4.4 wird deshalb ein Vorgehensmodell entwickelt, das auch diese personalen Aspekte berücksichtigt.

### Anwendungsfall Versicherungsplanspiel

Planspiele, die im Kontext der Versicherungsbetriebslehre angesiedelt sind, stellen spezifische Anforderungen an das zugrundeliegende Simulationsmodell:

– Wie jedes betriebswirtschaftliche Modell, so ist auch ein Versicherungsplanspiel von der Entwicklung der Modellsituation und den Auswirkungen von Entschei-dungen im

77 Der Aspekt der Dokumentation von Modelleigenschaften und -verhalten wird im Abschnitt 3.2 noch einmal detailliert aufgegriffen.

Zeitablauf geprägt. Viele Werte der Realität können nur als einzelne Ausprägungen und nicht als stetige Abbildung beobachtet werden. Wie bereits im Abschnitt 2.2 gezeigt wurde, fordert ein solches Planspiel als Basis ein diskretes, dynamisches Simulationsmodell.

– In einem Versicherungsplanspiel sollen Entscheidungen und deren Wirkungen über eine lange Zeit - etwa mehrere Jahre - hinweg beobachtet werden: Deshalb kommt nur eine Simulation im Zeitraffer in Frage.

– Zentraler Aspekt der Versicherung ist der Umgang mit Indeterminismus: Ein stochastisches Simulationsmodell bringt den Zufall ins Spiel.

Im folgenden soll der Fokus auf Planspiele mit betriebswirtschaftlichem Kontext und mit Spielunternehmen, die in einer Wettbewerbssituation agieren, gelegt werden. Wenn also von Planspielen die Rede ist, so seien damit immer Unternehmensplanspiele gemeint.

Der Kontext eines solchen Spiels wird in einem computergestützten Simulationsmodell repräsentiert. Dieses Modell bestimmt maßgeblich die Spiel- und Lernsituation und muß deshalb entsprechend den Anforderungen des Gesamtkonzeptes eines Planspiels entworfen und implementiert werden.

Die Modellbildung folgt dem ganzheitlichen Verständnis von Systemen. Um den Anforderungen von Unternehmensplanspielen gerecht zu werden, soll es als diskretes, dynamisches, stochastisches Zeitraffer-Simulationsmodell verstanden werden.

Im dritten Kapitel wird vor dem Hintergrund des Rahmens für Planspiele ein Konzept für anforderungsorientierte Planspielentwicklung erarbeitet. In den darauf folgenden Kapiteln vier und fünf wird dieses Konzept der objektorientierten Methodik folgend am Beispiel eines Versicherungsplanspiels umgesetzt.

„Wenn du ein Schiff bauen willst,
so trommle nicht Männer zusammen,
um Holz zu beschaffen, Werkzeuge vorzubereiten,
Aufgaben zu vergeben und die Arbeit einzuteilen,
sondern lehre die Männer die Sehnsucht
nach dem endlosen weiten Meer“[78]

# 3. Anforderungsorientierte Planspiele

Die im vorangegangenen Grundlagenkapitel herausgearbeiteten Ideen zum Lernen mit computergestützten Unternehmensplanspielen sollen im folgenden zu einem Konzept anforderungsorientierter Planspiele ausgebaut werden.

Für die Erarbeitung dieses Konzepts wurde eine zweistufige Vorgehensweise gewählt:

1. Der Einsatz von Planspielen als Instrument der Aus- und Weiterbildung legt den Grundstock für die Anforderungen an Planspiele.
2. Aufbauend auf den dort definierten Anforderungen wird ein Kriterienkatalog für die Konstruktion von Planspielen erarbeitet, der in der Folge als Basis für die Entwicklung und Beurteilung von Implementierungskonzepten verwendet wird.

## 3.1. Planspiele als Instrument der Aus- und Weiterbildung

Seit vielen Jahren wird das Instrument Planspiel mit wachsendem Anteil erfolgreich in der Aus- und Weiterbildung eingesetzt.[79] Verglichen mit konservativen Lehrmethoden wie Frontalunterricht, Vorträgen und Vorlesungen ist die Bedeutung des Planspiels in der Aus- und Weiterbildung aber eher gering.

Deshalb sollen die folgenden Ausführungen mit einem Blick auf neuere Ansätze der Lerntheorie beginnen - sie wird die Motivation für den Einsatz von Planspielen liefern. Im Anschluß daran werden praktische Erfahrungen und Forderungen der Lehr/Lernpraxis diskutiert.

78 Antoine de Saint-Exupéry

79 Vgl. Graf, J. (Planspiele, 1992), S. 6

Das Ziel ist es dabei, Konsequenzen und Forderungen für die Entwicklung von Planspielen herauszuarbeiten.

### 3.1.1. Lernen mit Planspielen

Mit Hilfe des Instruments Planspiel können vielfältige Aspekte des Lernens aufgegriffen und realisiert werden. Die folgende kurze Diskussion soll zeigen, wie neuere Ansätze der Lerntheorie und der Pädagogik mit Hilfe von Planspielen unterstützt werden können.

#### Allgemeine Überlegungen zum Lehren und Lernen

Bevor man sich Gedanken über anforderungsorientierte Planspiele machen kann, erscheint es sinnvoll, allgemeinen Gedanken über das Lehren und Lernen zu folgen.

*Vester* gibt hierzu eine Fülle wertvoller Anregungen. Er skizziert, wie seiner Meinung nach gelernt werden müßte, um den drängenden Problemen der Zukunft - wie z.B. erschöpfte Nahrungsreserven und Ressourcen, Überbevölkerung, Umweltverschmutzung etc. - erfolgreich begegnen zu können.[80] Zur Bewältigung der exponentiell fortschreitenden Informationsflut und der daraus resultierenden Stoffülle schlägt er den Einsatz alternativer Lernkonzepte vor. *Vester* spricht vom „Weg zu einer biologischen Lernstrategie“[81].

Er identifiziert die Hautprobleme des Lernens, die er in den Schwierigkeiten der Informationsvermittlung und den Problemen eines konventionellen, input/output-orientierten Lernprozesses begründet sieht:

- Lernen ist stark von Konventionen und Traditionen geprägt, die sich zum großen Teil noch nicht an den neueren Erkenntnissen des Lernens orientieren. Dies zeigt sich auch noch in den Bewertungs- und Belohnungssystemen und in der Organisation des Schul- und Lehrbetriebes.

- Die Halbwertszeit der Gültigkeit einmal vermittelten Fachwissens nimmt beständig ab.

- Die Lernenden müssen Fähigkeiten entwickeln, der Flut von neuen Erkenntnissen, Informationen und sich verändernden Rahmenbedingungen gewachsen zu sein.

- Die Wechselwirkungen von Körper und Geist wurden beim Lehren und Lernen bisher vernachlässigt. Dies manifestiert sich vor allem in der Vorstellung, Wissens-

80 Vgl. hierzu und zum folgenden: Vester, F. ( Neuland, 1981), S. 469 ff.; siehe auch ders. (Denken, 1978)

81 Vester, F. (Neuland, 1984), S. 469

vermittlung könne am besten alleine durch die Nutzung kognitiver Fähigkeiten des menschlichen Gehirns bewerkstelligt werden. Ausdruck dieser Vorgehensweise ist die Überbetonung der verbal-abstrakten Verarbeitung der Umwelt.

Richtet man seinen Blick auf die Situation an Schulen und Hochschulen, zeigen sich die Folgen der von *Vester* aufgeführen Problemfelder. Das zentrale Problem von Schul- und Hochschulabsolventen beim Eintritt in die Berufspraxis ist die mangelnde Anwendbarkeit gelernten Wissens, die sich wie folgt zeigt:[82]

1. Es wird der Gesamtheit von Sachverhalten und Daten zu wenig Aufmerksamkeit geschenkt.

2. Daten und Fakten werden unangemessen eng selektiert.

3. Kausalitäten können nicht identifiziert werden.[83]

### Alternative Lernkonzepte: Vernetztes Lernen

Vor dem Hintergrund dieser Probleme bieten sich Ansatzpunkte für alternative Lernkonzepte an, die *Vester* als „vernetztes Lernen“[84] bezeichnet:

- Lernen muß den Fähigkeiten des menschlichen Gehirns angepaßt werden, um dessen Fähigkeiten optimal nutzen können: *Vester* spricht von der anzustrebenden „Einheit von Körper und Geist“[85].

- Interdisziplinarität und die gemeinsame Arbeit von Lernenden mit unterschiedlichem Erlebnishintergrund bilden eine gute Basis, um vernetztes Denken zu erlernen und erleben.[86]

- Die Lernenden müssen den Umgang mit Wissen lernen und den Mut haben, die Speicherung auf Medien wie Buch oder Computer zu übertragen. Die Vermittlung

---

82 Vgl. Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993), S. 7 f.

83 Enge Selektion und fehlende Identifikation von Kausalitäten erscheinen auf den ersten Blick als Widerspruch - können doch innerhalb eines eng abgegrenzten Bereiches Kausalitäten in der Regel leichter entdeckt werden. Hier ist aber das Phänomen gemeint, daß globale Zusammenhänge und Kausalitäten durch enge Selektion nicht erfaßt werden können.

84 Vester, F. (Neuland, 1984), S. 475

85 Vester, F. (Neuland, 1984), S. 472

86 *Vester* führt eine Reihe von Versuchsprojekten auf, die mit integrativen Lernansätzen vernetzes Denken erleben lassen konnten. Sie erreichten erhebliche Effizienzsteigerungen gegenüber herkömmlichen Lehr- und Lernmethoden.
Vgl. Vester, F. (Neuland, 1984), S. 475 ff.

von Fähigkeiten zum kritischen Denken, zur Analyse und Synthese und zum Erkennen von Analogien müssen im Vordergrund stehen.

– Diese Fähigkeiten sind dann besonders gut zu nutzen, wenn nicht als Einzelkämpfer, sondern in der Gruppe gearbeitet wird. Deshalb gilt es, Lernmethoden zu favorisieren, die den Gruppenprozeß unterstützen.

## Alternative Lernkonzepte der Reformpädagogik

Die Gedanken *Vesters* zum Lernen spiegeln sich in den neueren Lernansätzen von Pädagogik und Psychologie wider.

### Konstruktivistischer Ansatz

Reformpädagogen korrigieren die traditionelle Vorstellung vom Lehren und Lernen, die bisher als direkte Wissensvermittlung vom Lehrer zum Schüler begriffen wurde.[87] Sie setzten dieser Vorstellung - aufbauend auf die Arbeiten von *Piaget*, *Dewey* und *Vygotsky* - die konstruktivistische Auffassung vom Lernen entgegen: Die Eigenaktivität des Lernenden in Form mentaler Konstruktionsprozesse wird als zentraler Aspekt gesehen.

*Mandl* / *Gruber* / *Renkl* stellen vor dem Hintergrund dieser Überzeugung Prinzipien zur Gestaltung von Lehr-/Lernumgebungen auf:[88]

1. Es muß an komplexen, authentischen Problemen gelernt werden. Das Problem sollte noch eingehender Problemdefinition bedürfen, um die Lernenden zur Erarbeitung von relevantem Wissen zu motivieren.

2. Durch die Verwendung multipler Perspektiven werden Kenntnisse und Fertigkeiten in verschiedenen Kontexten unter unterschiedlichen Zielsetzungen gelernt.

3. Die normalerweise intern ablaufenden Lern- und Problemlösungsprozesse sollen artikuliert werden. Durch die Offenlegung der Prozesse wird die Transferfähigkeit gefördert.

4. Es soll kooperativ gelernt und gearbeitet werden. Das produktive Austragen von Konflikten führt zu neuen Erkenntnissen.

---

87 Vgl. Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993), S. 8 f
Hier wird auch auf die einschlägige Literatur zur Reformpädagogik verwiesen.

88 Vgl. Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993), S. 9 f.

5. Lernen wird als Enkulturation verstanden: Neben Faktenwissen und spezifischen Fähigkeiten werden Denkmuster, Expertenkniffe, Überzeugungssysteme und ethische Standards erworben, die der bereits bestehenden Expertenkultur gleichen.

## Cognitive Apprenticeship

Eine Umsetzung dieser konstruktivistischen Denkweise findet sich in *Collins'* Modell des Cognitive Apprenticeship[89]: Authentische Aktivitäten und soziale Interaktionen führen den Lernenden in eine Expertenkultur ein.

Zentrum dieses Ansatzes ist ein am Konstruktivismus orientierter Handlungsrahmen zur Gestaltung der Lernumgebung. Es werden Inhalt (engl. content), Methoden (engl. methods), Abfolge des Lernens (engl. sequence) und Verhaltensaspekte (engl. sociology) der Lehr-/Lernsituationen beschrieben.

Im Hinblick auf das Instrument des Planspiels erscheinen einige Konkretisierungen der konstruktivistischen Grundannahmen bemerkenswert:

6. Es werden Techniken benötigt, die es den Lernenden ermöglichen, ihren Lernprozeß selbst zu überwachen und zu korrigieren. Hier hilft dem Lernenden ein Vergleich der Vorgehensweisen von Experten und Anfängern, indem die wichtigsten Schritte plakativ gegenübergestellt werden.

7. Die Lernenden sollen erfahren, daß ihr erworbenes Wissen in mehreren Kontexten angewendet werden kann: Dies gilt auch für so abstrakte **Lernziele** wie **Systemkompetenz** oder dem Einüben von Problemlösungsstrategien - Lernziele, die mit Planspielen besonders gut verfolgt werden können.

8. Situatives Lernen mit Kooperation, Wettbewerb und der Nutzung intrinsischer Motivation[90] mobilisiert in Gruppenarbeit sonst brachliegende Lernpotentiale.

9. Bei der Gestaltung dieser Lernumgebung müssen Zielkonflikte gelöst werden, die auch bei der Entwicklung von Unternehmensplanspielen Beachtung finden müssen:[91]

---

89 Vgl. Collins, A., Brown, S., Newman, S. (Cognitive Apprenticeship, 1989), S. 453 ff.
Ein kurzer Kommentar findet sich in: Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993), S. 10 f.

90 Unter intrinsischer Motivation versteht man immanente, aber verdeckte Motivatoren, wie beispielsweise die Motivation, sich in einer Situation profilieren zu wollen. Abzugrenzen davon sind extrinsische Motivationen, die klar erkennbar sind. Beispiele hierfür wäre etwa das Ziel, einen Titel, einen Schein oder eine Prämie zu erlangen. Intrinsische Motivation fördert die selbständige Aufgabenbearbeitung.
Vgl. Collins, A., Brown, S., Newman, S. (Cognitive Apprenticeship, 1989), S. 489

91 Siehe Collins, A. (Learning Environments)

Die folgende Tabelle zeigt für Inhalte und **Lernziele**, Lernumgebung und die Abfolge des Lernens stichpunktartig die relevanten Fragestellungen:

*Tabelle 5: Zielkonflikte bei der Gestaltung von Lernumgebungen*

| Kategorie | Zielkonflikte |
|---|---|
| Inhalte und Lernziele | Verständnis erarbeiten oder nur Wissen abrufen?<br>Erlernen ganzheitlicher Problemlösung oder nur der Lösung von abgegrenzten Teilaufgaben?<br>Vermittlung von Überblickswissen oder von Spezialwissen?<br>Verstehen der Zusammenhänge statt bloßem Zugriff auf Daten?<br>Erlangung geistiger oder körperlicher Fitness? |
| Lernumgebung | Interaktives, aktives oder passives Lernen?<br>Indirektes, fakultatives oder direktes Lernen?<br>Erlebnisorientiertes oder ernstes Lernen?<br>Natürliches oder effizienz-orientiertes Lernen?<br>Durch den Lernenden gesteuertes Lernen oder durch den Lehrer oder den Computer geführtes Lernen? |
| Abfolge des Lernens | Situationsbezogenes oder abstraktes Lernen?<br>Strukturiertes Lernen oder exploratives Lernen in schlechter strukturierten Lernumgebungen?<br>Lernen an systematisierten oder gemischten Aufgaben?<br>Lernen an einfachen oder komplexen Aufgaben? |

Eine Feststellung ist vor dem Hintergrund dieser umfangreichen Gestaltungsaufgaben notwendig: Detailliert zu diskutieren, wie diese Zielkonflikte für ein konkretes Planspiel gelöst werden können, würde den Rahmen der vorliegenden Arbeit sprengen - und ist auch nicht das Ziel. Wichtig ist in diesem Zusammenhang aber, daß eine Technik, die für sich in Anspruch nimmt, die Probleme der Planspielentwicklung anforderungsorientiert lösen zu können, die Problemgebiete und deren Implikationen kennt und Optionen für deren Bewältigung zur Verfügung stellt.

Blickt man auf die von *Collins* vorgeschlagenen Lehrmethoden[92], so lassen sich auch hieraus Ansatzpunkte für die Gestaltung von Planspielen ableiten:

| | |
|---|---|
| Lehrmethoden | Stufen abnehmender Intensität der Unterstützung der Lernenden bei der Aufgabenbearbeitung, wenn dies in kritischen Momenten gefordert ist:<br>- modelling (Modellierung)<br>- scaffolding (Unterstützung, Hilfestellung)<br>- coaching (Betreuung, schwächer als scaffolding)<br>- articulation (Artikulieren)<br>- reflection (Reflektion) |

*Tabelle 6: Lehrmethoden des Cognitive Apprenticeship-Ansatzes*

In Planspielen muß deshalb die Unterstützung für Teilnehmer und Spielleitung beispielsweise durch Werkzeuge, die die Interaktion mit dem Modell erlauben, ähnlich fein differenziert werden können.

Der folgende Abschnitt soll zeigen, daß sich aus den oben angeführten lerntheoretischen Überlegungen die wesentlichen Anforderungen in wenigen Aspekten konkretisieren lassen.

### Forderungen moderner Lernansätze an Planspiele

Können mit den in Planspielen verfügbaren Möglichkeiten die soeben ausgeführten Vorstellungen, Möglichkeiten und Forderungen modernen Lernens in die Praxis umgesetzt werden?

Damit die Frage beantwortet werden kann, soll nun versucht werden, aus den theoretischen Überlegungen zum Lernen Anforderungen an Unternehmensplanspiele abzuleiten.

92 Vgl. Collins, A., Brown, S., Newman, S. (Cognitive Apprenticeship, 1989), S. 480 ff.

## Forderungen aus der Aufgabe

Zentraler Aspekt moderner Lerntheorien ist die Arbeit an der Aufgabe. Der Schwerpunkt liegt auf Authentizität und **Komplexität**:

*Tabelle 7: Forderungen aus dem Kontext der Aufgabe an Planspiele*

| Lernen | Forderungen an Planspiele |
|---|---|
| Aufgabenbezogenes Lernen | Das Planspiel-Simulationsmodell muß ein vereinfachtes, aber in seinen Grundzügen getreues Abbild der Realität bieten.<br>Durch die Modellierung der wichtigsten Zusammenhänge und Schlüsselgrößen im Zeitablauf soll sich ein realistisches, vernetztes, dynamisches System ergeben, das dem Lernenden eine hinreichend komplexe, authentische Situation vermittelt.[93]<br>Das Modell soll multiple Perspektiven für eine Aufgabe bieten.<br>Der Lernende übt die Lösung von Aufgaben nicht an hypothetischen Fallbeispielen, er handelt authentisch in Situationen unterschiedlicher Kontexte, die der Realität sehr nahe kommen.[94] |
| Situationsadäquate Komplexität | Die Strukturiertheit und Komplexität der Aufgaben und Probleme muß den Vorkenntnissen der Lernenden und deren im Lernprozeß fortschreitenden Fähigkeiten Rechnung tragen.<br>Es muß dem Umstand Rechnung getragen werden, daß manche Aspekte auf eher abstrakten Niveau und andere sehr detailliert ausgearbeitet werden müssen.<br>Die Komplexität des Modells muß deshalb auch während des Lernprozesses variierbar sein. |

93 Effekte in dynamischen Systemen sind zum Beispiel Rückkoppelungen und carry-over-Effekte. Vgl. Günther, T. (Management, 1993), S. 25

94 ROHN, Initiator zahlreicher Aktivitäten zum Thema Planspiel, spricht deshalb von ‘Aktions-Lernen’: Rohn, W. (Didaktik, 1980), S. 7

## Forderungen aus den übergeordneten Lernzielen

Moderne Lernkonzepte legen ihren Schwerpunkt nicht so sehr auf die Vermittlung von abgegrenztem Fachwissen, sie betonen vielmehr die Bedeutung übergeordneter **Lernziele**:

*Tabelle 8: Forderungen übergeordneter Lernziele moderner Lerntheorien an Planspiele*

| Lernen | Forderungen an Planspiele |
|---|---|
| Lernen von vernetztem Denken | Es müssen Instrumente zur selbständigen Analyse des dynamischen Systemverhaltens vorhanden sein, die eine Überwachung und Korrektur des Lernprozesses ermöglichen.<br>Es soll dem Lernenden vernetztes Denken - also Analyse-, Synthese- und Kritikfähigkeit - vermittelt und von ihm gefordert werden:<br>Deshalb muß der Lernende von mechanischen Berechnungen und der reinen Speicherung von Daten und Fakten durch entsprechende Unterstützung (zum Beispiel durch den Computer) entlastet werden. |
| Einheit von Körper und Geist | Das Planspiel muß Lernen einer Spielsituation ermöglichen. So können intrinsische Potentiale des nichtkognitiven Lernens aktiviert werden.<br>Kurz gesagt: Die Mobilisierung des homo ludens. |
| Interdisziplinarität | Dem Plansspiel muß ein Systemmodell zu Grunde liegen, daß die Grenzen der klassischen Einteilung in Wissensgebiete und Disziplinen überschreitet. |

## Forderungen aus der Lehr/Lernumgebung

Moderne Lernkonzepte fordern interaktive Lehr/Lernumgebungen - schon auf den ersten Blick ein Domäne des Instruments Planspiel.

Die folgende Tabelle formuliert die Anforderungen detaillierter:

*Tabelle 9: Forderungen der Lehr/Lernumgebung moderner Lerntheorien an Planspiele*

| Lernen | Forderungen an Planspiele |
|---|---|
| Dokumentation der Inhalte des Modells | Es sind Instrumente zur Dokumentation der Inhalte des Modells, seiner Dynamik und seines Verhaltens für Konstrukteure, Spielleitung und Teilnehmer notwendig.<br>Diese Dokumentation muß in ihrer Interaktivität und ihrer Darstellung auf den Adressaten abgestimmt sein. |
| Gruppenarbeit | Gruppenarbeit muß unterstützt werden.<br>Teilnehmer sollen gemeinsam, im Wettwerb mit anderen ein Unternehmen führen und gemeinsam an Problemlösungen und Strategien arbeiten.[95] |
| Selbstgesteuertes Lernen | Die Teilnehmer müssen den Lernprozeß durch eigene Aktivitäten selbst steuern können. |
| Ermöglichung situationsadäquaten Eingreifens durch die Lehrenden | Die Spielleitung muß durch passende Werkzeuge bei der situationsadäquaten Dosierung ihrer Eingriffe in die Lernprozesse unterstützt werden:<br>Die Technik muß gerade an dieser Stelle ihren limitierenden Charakter verlieren. |

95 Vgl. Günther, T. (Management, 1993), S. 26
*Günther* betont die Heterogenität der Gruppen für die Effizienz von Planspielseminaren. Für Planspiele mit Führungskräften schlägt er interkulturell besetzte Gruppen vor, damit das inhärente Denken der Teilnehmer aufgebrochen werden kann.

## Lernsituationen für Planspiele

Die soeben skizzierten Anregungen neuerer Überlegungen zum Lernen können mit Hilfe des Planspiels in unterschiedlichen Lernsituationen realisiert werden:

*Tabelle 10: Lernsituationen für Planspiele*

| Lernsituation | Beschreibung |
|---|---|
| klassisches Planspielseminar[96]<br>mit Spielleitung | Der Spieltakt wird durch periodenorientiertes Arbeiten mit langen Bearbeitungszeiten vorgegeben.<br>Rhythmus von Entscheidungen und Auswertungen werden durch die Spielleitung bestimmt. |
| Fern-Planspiel<br>mit zentraler Koordination | Koordinations- und Kommunikationsprobleme erfordern längere Bearbeitungs- und Wartezeiten pro Periode.<br>Entscheidungen und Auswertungen werden durch den Spielleiter oder die Teilnehmer bestimmt.<br>Eine Planspielzentrale koordiniert den Spielablauf. |
| dynamisches Planspielseminar<br>mit Moderator<br>Teilnehmer greifen selbständig ein | Die Simulationszeit läuft während des Seminargeschehens ständig weiter.<br>Entscheidungen und Auswertungen können durch die Teilnehmer selbst vorgenommen werden. |
| Selbstlern-Planspiel<br>als Computer Based Training<br>ohne Fremdsteuerung | Der Teilnehmer bestimmt den Spielrhythmus selbst: Die Simulationszeit läuft während des Spiels ständig weiter und kann vom Lernenden kontrolliert werden.<br>Jede Systemkomponente und jedes Teilmodell kann einzeln gespielt werden.<br>Wirkungen und Sachzusammenhänge werden durch das Planspielprogramm auf Initiative des Spielers erklärt.<br>Szenariotechnik und Roll-back-Möglichkeit ermöglichen „Was wäre wenn“-Analysen durch den Teilnehmer.<br>Der Lernende spielt als Einzelkämpfer und kann daher die Vorteile der Gruppenarbeit nicht nutzen. |

96 Der Begriff des Seminars soll allgemein für eine Schulungsmaßnahme, die in logisch zusammenhängenden Blöcken durchgeführt wird, gebraucht werden.
Klassische Planspielseminare sind früher aus technischen Beschränkungen heraus nicht als geblockte Seminarveranstaltung durchgeführt worden. In der Praxis hat es sich jedoch gezeigt, daß eine geschlossene Seminarveranstaltung für den Lernerfolg sehr wichtig ist.

### Bemerkungen zu den Lernsituationen

Vom klassischen Seminar bis hin zur Selbstlernsituation steigen die Möglichkeiten des Planspiels für dynamisches, selbstgesteuertes Lernen.

Damit erhöhen sich aber auch die Anforderungen an die Flexibilität und den Funktionsumfang des Planspielmodells und die Mächtigkeit der verwendeten Werkzeuge für Konstruktion und Durchführung.

Ein Selbstlern-Planspiel fordert zwar technologisch die weitreichendsten Möglichkeiten an Dynamik, verzichtet jedoch auf die vielfältigen Aspekte des Lernens in der Gruppe. Computerunterstützte Selbstlernsysteme (engl. Computer Based Training, kurz: CBT) können deshalb im Sinne einer begleitenden Maßnahme, zum Beispiel für die Vor- oder Nachbearbeitung von Wissen, eingesetzt werden.[97] Diese Lernform soll deshalb im folgenden nur am Rande - als Weiterentwicklungsoption dynamischer Planspiele - betrachtet werden.

Wenn in der Literatur über Planspiele berichtet wird, handelt es sich fast ausschließlich um klassische Planspielseminare oder Fernplanspiele. Der Grund hierfür ist sicherlich in der fehlenden technischen Unterstützung und der mangelnden Erfahrung bei der dafür notwendigen mikroökonomisch fundierten Modellierung des Simulationsmodells zu suchen - letztgenannter Aspekt wird im Abschnit 4.3 noch eingehend diskutiert.

## 3.1.2. Planspiele als Instrument der Aus- und Weiterbildung an Schulen und Universitäten

Wie schon in den einführenden Bemerkungen zum Lernen ausgeführt wurde, so verändern sich auch die Anforderungen an die Ausbildung in Schulen und Universitäten: Es kann nicht mehr vorrangiges Ziel der Ausbildung sein, Fachwissen einzelner Bereiche zu vermitteln, vielmehr müssen die Lernenden in vernetztes Denken und systemorientierte Denkweise eingeführt werden.

### Planspiele in der Schule

Die Ergebnisse verschiedener Untersuchungen zum Betriebswirtschaftslehreunterricht zeigen Defizite bisheriger Lehrpläne und Unterrichtspraxis, die sich weitgehend mit den Erkenntnissen der Reformpädagogen (vgl. 3.1.1) decken.[98]

---

97 Vgl. Danzer, H. (CBT, 1994), S. 16f
*Danzer* greift ebenfalls den konstruktivistischen Ansatz des Congnitive Apprenticeship zur Gestaltung von CBT-Lernumgebungen auf.

98 Vgl. hierzu und zum folgenden: Achtenhagen, F. (Betriebswirtschaftslehreunterricht, 1992), S. 4 ff.

– Die Lehrinhalte bereiten die Schüler unzureichend auf die Anwendung des gelernten Wissens vor.

– Die Lernziele beziehen die Schüler zu wenig ein.

– Die Vermittlung von Inhalten geschieht vorzugsweise durch Frontalunterricht - eine Einbeziehung der Schüler findet nicht ausreichend statt.[99]

Die Auswertung der Ergebnisse mit dem Einsatz von Planspielen in Experimentalklassen ergab unter anderem folgende Erkenntnisse, die sich in der Planspielkonstruktion niederschlagen müssen:

– Neben der fachlichen Abstimmung von konventionell vermitteltem und durch Planspiele erlerntem Fachwissen ist die Unterstützung durch didaktisch geschulte Seminarleiter notwendige Voraussetzung für ein erfolgreiches Planspiel. Dazu gehören Erklärungen von Modellreaktionen, gezielte Hilfestellungen, Feedback zu Gruppenentscheidungen und der Kommentar zu vom Planspiel bereitgestellten Indikatoren.

– Kritische Situationen entstehen dann, wenn die Spieler glauben, die Planspielparameter zu kennen. Hinreichend komplexe Simulationsmodelle, die wie in der Realität einen Anteil an Indeterminismus beinhalten, können dem begegnen.

*Achtenhagen* schlägt nach den bisher gemachten Erfahrungen vor, eine Verbesserung der Lernsituation durch die Aufnahme von Planspielen in die Lehrpläne zu erreichen. Planspiele bieten seiner Ansicht nach als Prototyp komplexer Lehr-/Lern-Arrangements wesentliche Hilfen zu Gestaltung komplexer Lehr-/Lernprozesse.

### Planspiele an Universitäten

Mit steigenden Studentenzahlen an Hochschulen und Fachhochschulen und den immer lauter werdenden Forderungen nach verbessertem Praxisbezug der Hochschulausbildung gewinnt gerade im Zuge einer stärkeren Betonung der Lehre auch das Instrument des Planspiels zunehmend an Bedeutung.[100] Es soll hier keine generelle Diskussion über das Selbstverständnis der Hochschulen und der relativen Gewichtung von Forschung und Lehre geführt werden - vielmehr sollen die Möglichkeiten veränderter Lehr- und Lern-

99 Die Gründe für diese Situation sind vielfältig: Die Curricula schreiben zum Teil Lehrformen und -inhalte detailliert vor, die Kapazität der Lehrer reicht nicht aus, in der vorgegebenen Zeit schulungsintensive Lernformen zu verwenden, die Prüfungs- und Beurteilungssysteme sind nur der Vermittlung von Fachwissen angepaßt - sie werden zur Motivation der Schüler eingesetzt und auch von ihnen erwartet.

100 Vgl. Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993), S. 4 f.

formen und Anforderungen aus Sicht der Hochschule an das Instrument des Planspiels herausgearbeitet werden.

Anders als an Schulen, kann an Hochschulen sehr viel flexibler mit den Lehrprogrammen und -inhalten umgegangen werden. Planspiele haben deshalb schon seit langem Einzug in das Vorlesungs- und Übungsangebot gehalten.

Als ein typisches Beispiel sei hier das Integrierte Versicherungsplanspiel des INRIVER aufgeführt. Erfahrungen bei dessen Einsatz können zum Herausarbeiten von Anforderungen an die Planspielentwicklung genutzt werden.

## Das „Integrierte Versicherungsplanspiel“

Das INRIVER führt dieses Seminar seit 1987 durch. Basis dieses Planspieles ist das vom Großrechner auf PC portierte Planspiel-Simulationsprogramm VersPlan PC 1.0.

Das Versicherungsplanspiel hat sich in einem kontinuierlichen Prozeß der Weiterentwicklung seit 1989 zu einem viertägigen Blockseminar entwickelt. Es wird nun einmal im Jahr mit vier bis sechs Gruppen gespielt. Jedes dieser Teams besteht aus drei bis fünf Seminarteilnehmern:

- Studenten nach dem Vordiplom mit Schwerpunkt Versicherungen und ...
- Praktiker aus der Versicherungswirtschaft.

Die Seminarsituation ist die eines klassischen Planspielseminars:
Es werden sechs bis acht Spielperioden (die einem Geschäftsjahr entsprechen) durchgeführt, an deren Ende die Gruppen Entscheidungen für die nächste Spielperiode treffen. Die Gruppenmitglieder agieren als Vorstände ihres Versicherungsunternehmens. Diese Entscheidungen werden von den Teilnehmern in Formulare eingetragen und durch die Spielleitung in das Simulationsprogramm eingegeben. Die Teilnehmer werden nach erfolgter Simulation von der Spielleitung mit Unterlagen (in Papierform) über die Marktsituation und ihr eigenes Unternehmen versorgt.

## Überwindung von Designbeschränkungen

Die Beschränkungen des Designs von VersPlan PC 1.0 verhindern, daß individuelle Planspiel-Lernsituationen unterstützt werden. Gerade auch deshalb ist die Idee entstanden, mit einem völlig neuen Konzept eine anforderungsorientierte Planspiel-Entwicklungsumgebung zu entwerfen.

- Das Einstellen des Modells und seiner Parameter ist eine aufwendige und schwer zu kalkulierende Aufgabe[101]. Für ein weiterentwickeltes Planspiel muß dies bedeuten, daß mächtigere und transparentere Möglichkeiten und Hilfen zur zur Kalibrierung des Simulationsmodells gefunden werden müssen.

- Sollen inhaltliche Anpassungen (z.B. an veränderte Rechnungslegungsvorschriften) gemacht werden, sind umfangreiche, in ihrem Aufwand schwer abzuschätzende Maßnahmen nötig: Neuprogrammierung wesentlicher Teile des Modells und eine komplette Rekalibrierung wären nicht zu vermeiden. Es ist also praktisch nicht möglich, das Verhalten des Modells an die in den anderen (Pflicht-)Vorlesungen vermittelten Sachverhalte anzupassen, soweit dies nicht bereits zum Erstellungszeitpunkt der Fall war.

- Das Verändern der Komplexität des zugrundeliegenden Simulationsmodells ist nur in sehr engen Grenzen durch Weglassen einiger weniger Entscheidungen möglich. Es ist beim „Integrierten Versicherungsplanspiel“ nicht ohne aufwendige Neuprogrammierung möglich, die Komplexität (z.B. in Teilbereichen wie Kapitalanlage oder Außendienst) über das vorgegebene Maß zu erhöhen und an die Lernziele veränderter Teilnehmergruppen (z.B. Seminar mit oder ohne Praktiker, Haupt- oder Nebenfachstudenten) anzupassen.

- Das „integrierte Versicherungsplanspiel“ wurde bisher immer in der klassischen **Seminarsituation** gespielt: Weitergehende Konzepte, die die (Spiel-)Zeit weiter in das Seminargeschehen integrieren können, sind prinzipbedingt nicht möglich. Auch eine Vor-oder Nacharbeit der Inhalte mit Hilfe des gleichen Planspielmodells in einer Selbstlernsituation ist nicht durchführbar.

- Die heterogene Mischung der Arbeitsgruppen hat den Teamprozeß stark angeregt und Wissensasymmetrien auszugleichen geholfen. Die didaktisch richtig dosierte Unterstützung durch die Seminarleitung fördert diesen Prozeß. Aus Sicht der Modellierung ist zu beachten, daß Komplexität und Realitätsnähe ausreichen, um diese Möglichkeiten zu aktivieren.

### Forderungen an das Planspiel aus Sicht der Lehre

Planspielmodelle müssen den Anforderungen der speziellen Lehr-/Lernsituationen in Ausbildung und Lehre gerecht werden.

---

101 Zu *Hansens'* Integriertem Versicherungsplanspiel und den Weiterentwicklungen gibt es einige Studienarbeiten ( Islinger, G. (Schadensimulation, 1991), Haacke, V. (Analyse, 1989), Lamatsch, A. (Versicherungsplanspiel, 1984) ) in denen versucht wird, die Wirkzusammenhänge des Simulationsmodells zu dokumentieren und Empfehlungen für sinnvolle Gestaltung der Vorgabedaten und der Parameter zu finden.

Damit der Spielleiter die für den Planspielerfolg so maßgebliche Betreuung der Teilnehmer wahrnehmen kann, muß er vom Planspielprogramm soweit wie möglich bei den Aktionen unterstützt werden, die sich auf das Simulationsmodell beziehen:

- Der Spielleiter muß die Möglichkeit haben, das zugrundeliegende Modell in seiner Struktur und seinen Wirkzusammenhängen a priori zu erkennen. Anders als bei den meisten bisher verfügbaren Planspielen müssen Werkzeuge zur Exploration und Visualisierung von Struktur und Dynamik des Simulationsmodells zur Verfügung gestellt werden.

- Individuelle und standardisierte Auswertungen und Kennzahlen der Modellzustände im Zeitablauf ermöglichen dem Spielleiter, Modellreaktionen und Auswirkungen von Team-Entscheidungen zu jedem gewünschten Zeitpunkt zu erkennen und für die Teilnehmer transparent zu machen: Entsprechende Werkzeuge des Planspielprogramms müssen ihn dabei unterstützen.[102]

Planspiele können ihre Vorteile auch bei der Vermittlung von Fachkompetenz nur dann gezielt ausspielen, wenn eine Integration mit den anderen Lehrveranstaltungen gelingt: Das dem Planspiel zugrundeliegende Modell darf nicht durch seine Designbeschränkungen die Lehr-/Lern-Arrangements diktieren. Dies gilt umso mehr, je weniger sich die Teilnehmer untereinander durch unterschiedliche Vorkenntnisse in der Gruppe weiterhelfen können.

Der Aspekt der Vermittlung von Sozialkompetenz steht an Schulen und Universitäten immer noch im Schatten der Fachinhalte. Man muß jedoch bedenken, daß - durch die immer weiter zunehmende Menge an Faktenwissen - Fähigkeiten zu Teamarbeit, Arbeitsorganisation, Zeitmanagement, Informationsfilterung und Entscheiden in komplexen, vernetzten Systemen einen immer wichtigeren Stellenwert auch für Schüler und Studenten erlangen. So kann das Instrument Planspiel schon heute integrativ zur Vermittlung dieses Kompetenzbündels herangezogen werden. Ebenso dient es der Erleichterung des Einstiegs in die Berufspraxis, da gerade diese Faktoren den ‘Praxisschock’ hervorrufen.[103]

*Rohn* betont als wichtigsten Aspekt bei der Anwendung des Planspiels als Ausbildungsinstrument in der Lehre die Möglichkeit, theoretisch erworbenes Wissen durch „Aktionslehre“[104] im Planspiel anzuwenden. Das Planspiel bietet für die Studenten eine der wenigen Möglichkeiten, den Bogen vom theoretisch erworbenen Wissen zum Management-Handeln in der Praxis zu verwirklichen. Für das Planspielmodell bedeutet dies, daß realitätsnahe Eingriffsmöglichkeiten und Reaktionen realisiert werden müssen.

---

102 Dies fordert auch ROHN in: Rohn, W. (Didaktik, 1980), S. 48

103 Vgl. Heidack, C. (Lerninstrument, 1992), S. 56f

104 Rohn, W. (Didaktik, 1980), S. 61

### 3.1.3. Planspieleinsatz in der betrieblichen Aus- und Weiterbildung für Führungskräfte

Die Verschärfung des internationalen Wettbewerbs zwingt zu einer Professionalisierung des Managements: Eine entsprechende Führungskräfte-Weiterbildung muß diesem Trend Rechnung tragen. Es werden geeignete Lehr- und Lernmethoden benötigt, die die Managementaufgaben realitätsnah unterstützen.[105]

#### Motivation für den Einsatz von Planspielen

Das Instrument des Unternehmensplanspiels bietet sich gerade für den Einsatz als Instrument zur Führungskräfteweiterbildung an:

- Statt des traditionellen fallbasierten Lernens kann während eines Planspiels Management-Handeln struktur- und ablaufbezogen geübt werden.

- Den Teilnehmern wird sanktionsfrei ermöglicht, Strategien zu erproben und deren Wirkungen und Zusammenhänge zu erkennen. Dabei ist es auch möglich, unkonventionelle Strategien zu testen.

- Es werden dem Lernenden Situationen geboten, wie sie auch real vorkommen können.

- Zwei Aussagen *Rohns* charakterisieren den Einsatz von Planspielen für die Führungskräfteweiterbildung: „Entscheidung ist der Kern des Führens“ [106]: Deshalb wird in Planspielen „Führungslernen als Entscheidungsübung“ [107] praktiziert.

- Planspiele zählen zu den Formen der Weiterbildung, bei denen die Teilnehmer selbst das größte Engagement zeigen: Sie übernehmen Verantwortung für ihr Unternehmen und werden durch den Wettbewerbsdruck zusätzlich motiviert.[108]

#### Forderungen an das Planspielmodell

Beim Entwurf eines Planspielmodells für diesen Einsatzzweck steht deshalb die Forderung nach einem möglichst realistischen Abbild der Realität im Vordergrund:

- Die im vorigen Abschnitt über den Einsatz von Planspielen in Ausbildung und Lehre angesprochenen Anforderungen treffen im Prinzip auch für den Einsatz in der Führungskräfteschulung zu. Schwerpunkt wird jedoch hier auf die Rück-

105 Vgl. Rohn, W. (Praxis, 1992), S. 20 ff.

106 Rohn, W. (Führungsentscheidung, 1964), S. 15

107 Rohn, W. (Führungsentscheidung, 1964), S. 15

108 Vgl. Günther, T. (Management, 1993), S. 27

koppelungseffekte mit der bisherigen Berufspraxis und der Verbesserung der Sozialkompetenz liegen.

- Die Wirkzusammenhänge des Modells müssen die in der Praxis beobachteten Reaktionen hervorrufen und bei extremen Situationen noch zu nachvollziehbaren Ergebnissen führen.

- Die Entscheidungen müssen in ihren wesentlichen Elementen den Aktionsmöglichkeiten in der realen Entscheidungssituation entsprechen.

- Die Komplexität des Modells muß hinreichend groß sein, da die Teilnehmer bereits über eine große Erfahrung verfügen.

## 3.2. Entwicklung eines Kriterienrahmens für anforderungsorientierte Planspielentwicklung

Im vorhergehenden Abschnitt wurden die Anknüpfungspunkte für den Einsatz von Planspielen in der Aus- und Weiterbildung detailliert dargestellt.

Im folgenden wird mit der zusammenfassenden Darstellung der Anforderungen ein Instrumentarium für deren Umsetzung in Planspielen erarbeitet.

Dafür wurde eine zweistufige Vorgehensweise gewählt:

1. Im ersten Schritt wird ein Anforderungskatalog an Planspiele, dessen Simulationsmodell und Werkzeuge erarbeitet.

2. Diese Anforderungen werden in einem zweiten Schritt in Anforderungen an die Implementierung des Planspielmodells, der Werkzeuge und deren systemtechnische Basis konkretisiert.

### 3.2.1. Anforderungen an Planspiele

Aus den im Abschnitt 3.1 formulierten Anforderungen an Planspiele lassen sich drei Fragestellungen zu Anforderungen an das Planspiel-Simulationsmodell und die unterstützenden Werkzeuge ableiten:

1. Welche Anforderungen folgen aus der Abbildung des relevanten Wirklichkeitsausschnittes?

2. Welchen Konsequenzen hat die Anforderung der Unterstützung unterschiedlicher Lehr-/Lernsituationen für das Simulationsmodell und die Werkzeuge?

3. Wie kann das Modell dokumentiert und damit kommuniziert werden?

## Die Mikrowelt - der Wirklichkeitsausschnitt im Simulationsmodell

Zentrum eines computerunterstützten Unternehmensplanspiels ist die Abbildung eines relevanten Wirklichkeitsausschnittes - verstanden als System - in einem Simulationsmodell.

Da im Planspiel der Lernende im Mittelpunkt des Interesses steht, muß sich die Modellierung auch maßgeblich an ihm orientieren:

*Tabelle 11: Die Mikrowelt im Planspiel-Simulationsmodell*

| Kriterium | Ausprägung | Anforderung an das Simulationsmodell |
|---|---|---|
| Wirklichkeits ausschnitt (System) | Kontext | Realitätsnähe und Plausibilität der Eingriffs-möglichkeiten und Reaktionen als Grundlage für Authentizität müssen bei der Modellierung an erster Stelle stehen. |
| | Komplexität | Die Komplexität des Modells entsteht durch Vernetzung der Systemkomponenten zu einem dynamischen Systemmodell: Die Komplexität muß je nach Bedeutung variierbar sein. |
| | Veränderlich-keit | Damit die Modellierung veränderlicher Wirklich-keitsausschnitte möglich ist, muß das Simula-tionsmodell leicht veränderbar und robust gegenüber Erweiterungen und Veränderungen sein. |
| Lernender | Erfahrungs-stand | Die Komplexität und die Reaktionen des Modells müssen an die Vorkenntnisse und den im Laufe des Lernprozesses sich verändernden Erfahrungsstand der Lernenden anpaßbar sein. |
| | Erwartungen | Die Lernenden erwarten von der Modellierung Realitätsnähe - erst dann wird authentisches Verhalten (anstatt eines Rollenspiels) die Motivation und den Lernerfolg erhöhen. |

### Unterstützung verschiedener Lehr-/Lernsituationen

Wie bereits im zweiten Kapitel ausgeführt, sollte ein Planspiel flexibel genug sein, um den unterschiedlichen Anforderungen verschiedener Lehr-/Lernsituationen gerecht zu werden.

Die folgende Tabelle zeigt, wie sich dies in Anforderungen an das Simulationsmodell niederschlägt:

*Tabelle 12: Unterstützung verschiedener Lehr-/Lernsituationen*

| Kriterium | Ausprägung | Anforderung an das Simulationsmodell |
|---|---|---|
| Seminar-situation | klassische Situation | Periodenorientierung<br>Auswertungen durch Spielleitung |
| | Fern-Planspiel | siehe oben |
| | dynamisch | Die Simulationszeit läuft während des Spiels - die Zustände der Systembestandteile ändern sich fortwährend.<br>Die Teilnehmer müssen laufend die Möglichkeit haben, Informationen über Systemzustände und Änderungen zu erhalten. |
| | selbst-lernen | Szenarien zeigen Konsequenzen alternativer Entscheidungen und Situationen |
| Verhaltens-aspekte | Gruppenarbeit | Entscheidungen und Auswertungen müssen (auch) auf Gruppenbasis möglich sein.<br>Besonders bei Spielen, die von den Teilnehmern direkt am Computer gespielt werden können, muß die Gruppenzugehörigkeit im Modell repräsentiert werden. |
| | intra-Gruppen-Kommunikation | Die Modellierung muß Raum für Aufgaben bieten, die durch Aktionen zwischen den Gruppen bearbeitet werden. [109] |
| | Steuerung der Problemfelder | Problematisierung oder Entschärfung der Aufgabe für die Teilnehmer durch die Spielleitung muß robust und kontrollierbar möglich sein. |

### Anforderungen an die Dokumentation des Modells

Ein Aspekt, der leicht von der Beschäftigung mit Lernzielen und der Modellbildung in den Hintergrund gedrängt wird, ist die Forderung nach Dokumentation des Modells: Nur, wenn Inhalte auch kommuniziert werden können, kann ein Erkenntnisfortschritt aus

109 Die Kommunikation zwischen den Teilnehmern könnte durch elektronische Post (engl. electronic mail) unterstützt werden: Da electronic mail mittlerweile flächendeckend in Unternehmen Einzug gehalten hat, könnte eine äquivalente Funktion die Authentizität im Planspiel erhöhen.

ihnen gezogen werden.[110] Dies bezieht sich auf die Dokumentation für den Modellkonstukteur und seinen Auftraggeber, die Spielleitung und die Teilnehmer eines Planspiels.

Folgende Tabelle greift diese Anforderung auf und gliedert sie nach den Zielgruppen[111]:

*Tabelle 13: Dokumentation des Modells*

| Dokumentation für ... | für welchen Zweck? | Anforderungen an das Modell und die Werkzeuge |
|---|---|---|
| den Modell-Konstrukteur | Konstruktion | Möglichkeiten zur Erschließung der Bedeutung vorgefertigter Teile |
| | Validierung | Testmöglichkeit und Dokumentation von Teilmodellen und des Gesamtmodells |
| | Dokumentation | Werkzeuge zur Erstellung von Modellbeschreibungen für Spielleiter und Teilnehmer |
| Spielleiter | Vorbereitung | Reduzierung der **Komplexität** des Modells<br>Werkzeuge zum Erschließen und Kalibrieren des Modellverhaltens und der Struktur |
| | Unterstützung der Lernprozesse | Werkzeuge zur Offenlegung der Prozesse |
| Teilnehmer | Kontrolle des Lernprozesses | Interaktive Werkzeuge zur Offenlegung sowie Erklärung von Systemeigenschaften und -verhalten.<br>Möglichkeiten zur Dokumentation von Prozessen, Ursachen und Wirkungen müssen vorhanden sein. |

Die Anforderungen an Modell und Werkzeuge sind für die verschiedenen Zielgruppen sehr ähnlich: Ein Unterschied besteht in der abweichenden Betonung der Interaktivität, der Tiefe des Eingriffs in das Simulationsmodell und des Entwicklungsprozesses.

110 *Freiburghaus* stellt deshalb diesen Aspekt bei der Entwicklung einer objektorientierten Simulationsmethodik sogar in den Vordergrund seiner Betrachtungen.
Vgl. Freiburghaus, M. (Simulation betriebswirtschaftlicher Systeme, 1993), S. 60 ff.

111 Unter Zielgruppen für ein Planspiel sollen im folgenden diejenigen verstanden werden, die dem Umfeld der Planspielentwicklung und -durchführung zuzuordnen sind: Konstrukteure (Ersteller) und Schulungsverantwortliche (Auftraggeber), Spielleiter und Teilnehmer.

### 3.2.2. Anforderungen an die Implementierung von Planspielen

Die im vorangegangen Abschnitt aufgestellten Anforderungen an das Simulationsmodell und die Werkzeuge, die bei Entwicklung und Durchführung von Unternehmensplanspielen benötigt werden, haben Konsequenzen für deren Implementierung.

Deshalb sollen im folgenden hieraus Anforderungen an die Implementierung formuliert werden. Als wesentliche Problemfelder lassen sich herausarbeiten:

1. Über die Beschäftigung mit dem Thema **Komplexität** gelingt die Integration des fachlich-inhaltlichen Aspektes (dem Kontext) mit den lerntheoretischen Überlegungen des Konstruktivismus: Authentische Simulationsmodelle bilden über situationsadäquate Komplexität die Basis für aufgabenbezogenes, interaktives, selbstgesteuertes Lernen.

2. Realisiert werden kann dies in verschiedenen Seminarsituationen, die unterschiedliche Anforderungen an die Implementierung stellen.

#### Komplexität, Realitätsnähe und Authentizität

Wie bereits angesprochen, sind richtig gewählte Komplexität, Realitätsnähe und Authentizität drei wesentliche Merkmale, die anwendungsbestimmte Planspiele kennzeichnen. Doch diese Ziele zeigen nur scheinbar in die gleiche Richtung.

#### Komplexität von Realität und Modell

> Die Realität hat mit ihrer für menschliche Vorstellungskraft undurchdringbaren Komplexität und Vernetzung dazu geführt, Erkenntnisse aus der Beschäftigung mit Teilsystemen zu ziehen.
>
> Je nachdem, wie diese Teilsysteme abgegrenzt werden, übersieht man aber die Wechselwirkungen dieser Systeme untereinander: *Vester* bezeichnet diese Vorgehensweise als technokratisches, unsystemisches Denken[112]. In der Ausbildung führt dies zu Spezialisten, die alle Details eines Fachgebietes beherrschen. Detaillierte Erkenntnisse in scheinbar unabhängigen Systemen führen jedoch zu Entscheidungen und Eingriffen, die im globaleren Zusammenhang kontraproduktive Wirkungen erzielen können. Die systemische Denkweise hingegen bezieht die Wechselwirkungen der Systembestandteile in die Betrachtung ein: Teilsysteme (wie z.B. der Innendienst eines Versicherungsunternehmens) werden nur mit wenigen, charakterisierenden Objekten modelliert: Es erscheint auf den ersten Blick also als wenig komplex; Fachspezialisten mögen es gar als trivial oder realitätsfern

112 Siehe Vester, F. (Neuland, 1991)

bezeichnen. Dennoch erreicht ein Modell über die Vernetzung mäßig komplexer Teilsysteme ein beachtliches Maß an Komplexität und damit letztendlich auch an Realitätsnähe.

Als Kritikpunkt an der Realitätsnähe der Modellierungen in Simulationen und Planspielen ist immer die fehlende Verhaltenskomponente des Modells angeführt worden:[113] Doch müssen personale Aspekte wirklich in einem Simulationsmodell eines Planspiels implementiert werden?

Diese Frage führt zur Diskussion der Begriffe Realitätsnähe und Authentizität. Authentizität kann als mehr aufgefaßt werden als 'Handeln in Situationen, in denen Fachaufgaben bewältigt werden müssen, die denen in der Realität gleichen':

- Authentisches Verhalten kann auch in einer Spielsituation entstehen, die auf einem Modell aufbaut, das selbst keine Verhaltensaspekte implementiert - also auf den ersten Blick unrealistisch erscheint. Allein durch die zwischenmenschlichen Aspekte bei der Beschäftigung mit der Spielsituation entsteht ein hohes Maß an Intensität - und damit auch an Authentizität.[114]

- Doch dies bedeutet natürlich nicht, daß der Grad der Übereinstimmung von Realität und Modell für den Planspielerfolg unwichtig ist. Die Vermittlung von Fachkompetenz und die Integration mit bereits an anderer Stelle gelerntem Wissen steigt mit dem Deckungsgrad im Planspielmodell.

### Forderungen aus der Komplexität des Modells an die Implementierung

Damit Planspielteilnehmer aus den Wirkungen ihrer und anderer Entscheidungen lernen können, muß das Planspiel auch adäquate Möglichkeiten zur Dokumentation von komplexen Zusammenhängen bieten:

Modellverhalten, das zumindest in seinen grundlegenden Aspekten dem Verhalten in der Realität entsprechen muß, stellt die Basis für die Nachvollziehbarkeit von Entscheidungen dar: Denn nur so können die Teilenehmer bereits gelernte oder erfahrene Sachverhalte wiederfinden, bestätigen, ergänzen und vertiefen.

Diese Vermittlung von Modellreaktionen an den Spielleiter und an die Teilnehmer sollte durch Erklärungen des Modellverhaltens zusätzlich unterstützt werden.

---

113 Vgl. Heidack, C. (Lerninstrument, 1992), S. 50ff.

114 Diese Erkenntnis führt zu einer stärkeren Betonung des Trainings von Verhaltensaspekten im Planspiel.
Vgl. Schmidt, H. (Schlüsselqualifikation, 1994), S. 1ff.

### Offenlegung der Systemzusammenhänge

Ein erster Schritt zur Erklärung ist die Vermittlung der Systemzusammenhänge durch Visualisierungen. Gut geeinget sind hierfür graphische Darstellungen, die es dem Benutzer ermöglichen, relevante Informationen auf verschiedenen Abstraktionsebenen zu erhalten. Dies kann interaktiv - durch Benutzung eines entsprechenden Werkzeugs - oder einseitig durch gedruckte Berichte erfolgen.

Voraussetzung für ein solches Vorgehen ist die Komposition eines hierarchischen Modells aus rekursiven Teilsystemen, wie es im zweiten Abschnitt über Systeme beschrieben wurde.

- Graphen können Informationen über die Systembestandteile und deren Verknüpfungen besser darstellen, als dies textuelle Darstellung allein könnte. Durch die Anzeige von Systemen verschiedener Aggregationsstufen, kann die Komplexität der jeweils betrachteten Systeme erheblich reduziert werden:
  - Auf der Ebene hochaggregierter Teilsysteme erhält man einen Überblick über die Systemstruktur im Großen.
  - Auf der Ebene detaillierterer Teilsysteme kann Systemaufbau und Systemverhalten im Kleinen untersucht werden.
- Die Interaktion der Systembestandteile während der Simulation kann in Ereignislisten protokolliert werden. Sind diese Ereignisse selbst als Objekte implementiert, kann ein Spielleiter oder Spielkonstrukteur die lokalen Aktionen direkt nachvollziehen.

### Offenlegung der Systemdynamik

Ein wesentlicher Aspekt des Systemverhaltens manifestiert sich in dessen Systemdynamik:

- Unter dynamischem Verhalten von Simulationsmodellen wird in erster Linie die Veränderung der Eigenschaften der Systembestandteile - also etwa der Schlüsselvariablen des Modells - verstanden. Planspielentwicklung muß für die Möglichkeiten sorgen, diese Entwicklungen zu verfolgen, Kennzahlen zu bilden und einander gegenüber zu stellen. Die Auffassung von Dynamik reicht dabei von der qualitativen Dynamik, die den Augenmerk auf die logischen Zusammenhänge während einer Periode legt, bis zur quantitativen Dynamik, die den Faktor der Zeit in die Betrachtungen einbezieht (vgl. 2.2)
- Doch es kommt noch ein weiterer Aspekt der Dynamik hinzu: Das Modell kann sich im Laufe der Zeit in seiner Struktur selbst verändern. Die Visualisierung

von Strukturänderungen erfordert die dynamische Erstellung von entsprechenden Graphen und Berichten.

### Nutzung der Simulation zur Entscheidungsvorbereitung

Jedem computerunterstützten Unternehmensplanspiel liegt ein Simulationmodell zu Grunde. Was liegt also näher, als die Möglichkeiten der Simulation auch für die Vorbereitung von Entscheidungen zu benutzen.

Im Falle eines Versicherungsplanspiels könnte zum Beispiel die simulative Berechnung der Gesamtschadenverteilung eine Unterstützung zur Prognose des Schadenaufkommens und damit letztendlich zur Prämiengestaltung bieten.

Dieser Aspekt gewinnt besonders dann an Bedeutung, wenn in einer Selbstlernsituation gearbeit wird. Neben der Simulation sind auch noch andere Entscheidungsunterstützungen, wie zum Beispiel ‘intelligente’ Assistenten zur Erstellung von Auswertungen und Berichten oder Ratgeber zur direkten Entscheidungsunterstützung[115] denkbar.

115 Das Planspiel CABS, ein Planspiel über europäische Automobilproduzenten, zeigt genau in diese Richtung: Hier heißen die Ratgeber zur Unterstützung von Selbstlernsituationen und zur Hilfe für Einsteiger „virtuelle Manager“. Vgl. Virtual Management (Cabs, 1994), S. 10 ff.

### Unterstützung verschiedener Seminarsituationen

Die Lernsituation bestimmt maßgeblich die Anforderungen an Modell und Werkzeuge. Die folgende Tabelle greift die Forderungen an die Modellierung auf und leitet Anforderungen an die Implementierung ab:

*Tabelle 14: Lernsituationen und deren Anforderungen an die Planspielentwicklung*

| Lernsituation | Anforderungen an das Planspielmodell | Anforderungen an die Implemetierung |
|---|---|---|
| klassisches Planspielseminar mit Seminarleitung | Periodenorientiertes Arbeiten<br>Entscheidungen und Auswertungen durch den Spielleiter<br>Anpassung von Kontext und Komplexität an das Lernziel<br>Nachvollziehbarkeit der Wirkungen eigener und fremder Entscheidungen | Werkzeuge und Komponenten zur Konstruktion von Modellen variabler Komplexität<br>Unterstützung der Spielleitung bei der Modellkalibrierung<br>Schnelle Durchführung der Simulation und Erstellung der Auswertungen |
| Fern-Planspiel mit Seminarleitung | Periodenorientiertes Arbeiten<br>Entscheidungen und Auswertungen durch Spielleiter oder Teilnehmer | Unterstützung langer Transaktionen bei „online-Spiel“ |
| dynamisches Planspielseminar<br>mit Seminarleitung Teilnehmer greifen selbständig ein | Simulationszeit läuft während des Seminars ständig weiter<br>Entscheidungen und Auswertungen durch Teilnehmer | Mehrbenutzerfähigkeit<br>Flexible, leicht zu bedienende, interaktive Werkzeuge zur Eingabe von Entscheidungen und für Auswertungen |
| Selbst lernen Computer Based Training | Erklärung der Wirkungen und der Sachzusammenhänge<br>„Was wäre wenn“ Analysen | Werkzeuge zur Erklärung der Wirkzusammenhänge<br>automatisierte Spielpartner |

### Konsequenzen für die Planspielentwicklung

Vom klassischen Seminar bis zur Selbstlernsituation steigen die Anforderungen an das Planspielmodell und die Werkzeuge immer mehr an.

– Ein überzeugender und zukunftsorientierter Entwurf einer Planspiel-Entwicklungsumgebung sollte variable, möglichst aufwärtskompatible Lösungen für verschiedene Seminarsituationen anbieten können.

– Eine Variation der Seminarsituation je nach Teilnehmer und Lernzielen sollte durch das verwendete Planspielprogramm nicht verhindert werden.

## Warum immer klassische Seminarsituation?

Bisher wurden computerunterstützte Unternehmensplanspiele überwiegend in Form eines klassischen Planspielseminars oder eines Fernplanspiels durchgeführt, da die gängigen Planspiel-Implementierungen diese beiden Seminarformen unterstützen.[116]

Schwerpunkt bei der Weiterentwicklung von Planspielen waren bisher die Bemühungen zur Nutzung der Potentiale der in der Spielsituation auftretenden Verhaltensaspekte für die Vermittlung sozialer Kompetenz. Es wurde darüber nachgedacht, wie man den pädagogischen Rahmen so an die Möglichkeiten vorhandener Planspiele anpassen kann, daß größtmöglicher Nutzen daraus zu ziehen ist. Daß man den offensichtlich bestehenden Anpassungsproblemen aber auch mit einer verbesserten technischen Basis begegnen kann, kommt erst seit kurzem zur Diskussion.

Wegen fehlender technischer Möglichkeiten besteht bisher ein großes Defizit an weitergehenden, pädagogisch fundierten Seminarkonzepten.

116 Vgl Graf, J. (Marktübersicht, 1992), S. 95 ff.

# 4. Einsatz des objektorientierten Paradigmas zur Entwicklung von Unternehmensplanspielen

Das vierte Kapitel widmet sich detailliert dem objektorientierten Ansatz und seinem Einsatz für die Entwicklung von Unternehmensplanspielen.

1. Zuerst soll mit der Vorstellung der Prinzipien, Methoden und Techniken objektorientierter Systeme die Grundlage für die weitere Vorgehensweise gelegt werden.

2. Dann werden die Möglichkeiten des objektorientierten Paradigmas für die Planspielentwicklung im Grundsatz diskutiert.

3. Der dritte Abschnitt widmet sich den Beiträgen der Objektorientierung für Lösungsansätze zur Bewältigung der Problemdimensionen der Planspielentwicklung.

4. Den Abschluß bildet die Entwicklung eines Vorgehensmodells für objektorientierte Planspielentwicklung.

## 4.1. Objektorientierte Systeme

Bevor über den Einsatz des objektorientierten Paradigmas für die Planspielentwicklung diskutiert werden kann, müssen zuerst die grundlegenden Konzepte der Objektorientierung vorgestellt werden.

Objektorientierte Modellierung bietet die Methodik für problemnahe Analyse, Modellbildung und Implementierung, objektorientierte Programmiersprachen liefern die Unterstützung für die Implementierung der Methodik und objektorientierte Datenbanken wenden Prinzipien der objektorientierten Programmierung an und erweitern sie um Konzepte der Datenbanktechnik.

### 4.1.1. Objektorientierte Modellierung

Am Anfang steht eine triviale, aber leider nicht immer als selbstverständlich akzeptierte Aussage: Ein Datenverarbeitungssystem - bestehend aus Hardware und Software - ist nicht Selbstzweck, sondern Lösung für ein Problem.

### Modell als Abbild der Realität

Diese Lösung bildet die Realität als ein auf die wesentlichen Aspekte reduziertes Modell ab. Damit aber komplexe Probleme in Software abgebildet werden können, müssen die Aufgaben in Teilgebiete zerlegt werden. Der naheliegenden Abgrenzung von *Frauenstein* / *Pape* / *Wagner*[117] folgend, kann dabei die klassische (pragmatische, technologische) von der systemorientierten (objektorientierten) Denkweise unterschieden werden:

### Die klassische Denkweise

Klassische Denkweise der in der Datenverarbeitung beheimateten Fachleute war es bisher, Funktionen und Daten der Realität zu erkennen und modular strukturiert darzustellen. Man spricht von strukturierter Analyse und Design und prozeduraler Programmierung. Werkzeuge dieser Denkweise sind CASE (Computer Aided Software Engineering)-Tools und Programmiersprachen der dritten Generation wie COBOL oder C.

Legt man diese traditionelle Denkweise jedoch einem nicht durch die in der üblichen EDV-Erziehung vorbelasteten Fachmann[118] für ein Aufgabengebiet vor, wird man feststellen, daß eine Modellierung des diesem Experten vertrauten Problems ganz anders aussehen würde: Es sind Dokumente, Vorgänge, Bearbeiter, Produkte, Kunden oder ähnliches an dem Problem beteiligt. Ein erster Modellierungsversuch wird sich auch zuerst mit diesen Objekten beschäftigen - und nicht mit Datenstrukturen, Funktionen und Modulen.

### Die systemorientierte, objektorientierte Denkweise

Folgt man der systemorientierten Denkweise, ist Modellierung die Abbildung eines Systems und seines zeitlichen Verhaltens in ein abstraktes Modell. Zeitliches Verhalten entspricht dem Informationsprozeß des betrachteten Wirklichkeitsauschnittes.

- Ein Modell ist die Beschreibung einer Ansammlung von Phänomenen mit Begriffen.

- Abstraktion findet statt, um für Phänomene der Wirklichkeit neue, für den Zweck des Modells taugliche Begriffe zu finden:

    - Klassifikation transformiert die Menge der Phänomene in die Menge der Begriffe. Exemplifikation bedeutet die inverse Abbildung.

---

117 Vgl. hierzu und zur systemorientierten Denkweise: Frauenstein T., Pape U., Wagner O. (Sprachkonzepte, 1990), S. 8 ff.

118 Gibt es so etwas wie einen ‘unbelasteten Fachmann’ eigentlich? Die prozedurale Denkweise der bisher etablierten EDV (angewendet auf Aufgaben, wo sie eigentlich unangebracht ist) hat uns eigentlich alle spätestens im Laufe der Ausbildung geprägt.
Vgl. Freiburghaus, M. (Simulation betriebswirtschaftlicher Systeme, 1993), S. 50

- Durch Komposition entstehen neue Begriffe mittels Anordnung vorhandener Begriffe.

- Generalisierung und Spezialisierung führen zu einer Klassifikationshierarchie der Begriffe.

Das objektorientierte Paradigma folgt dieser systemorientierten Vorstellung von Abstraktion sehr natürlich: Es modelliert die Realität aus Objekten und den Beziehungen zwischen diesen Objekten.

Objekte der Realität haben Eigenschaften und Fähigkeiten[119]: In objektorientierter Terminologie wird dies direkt übernommen und heißt dort properties und methods.[120] Den Zustand eines Objekts bestimmen die aktuellen Eigenschaften und die Aktionen auf dieses Objekt.

Sieht man sich die Objekte näher an, wird man feststellen, daß sie über Gemeinsamkeiten verfügen. Es gibt Objekte, von denen es gleichartige gibt, Objekte die sich z.B. nur in der Ausprägung ihrer Eigenschaften unterscheiden. Man denke nur an verschiedene Mitarbeiter in der Buchhaltung. In der objektorientierten Methodik werden deshalb allgemeine Beschreibungen von Objekten (z.B. Buchhalter) modelliert, die dann Vorbild für die Erzeugung der individuellen Objekte sind. Allgemeine Beschreibungen speichert man in objektorientierten Programmiersprachen in Klassen, individuelle Objekte sind Exemplare[121] dieser Klassen.

Doch die Gemeinsamkeiten gehen noch weiter. So sind z.B. die Klassen Buchhalter und Kalkulatoren beides Mitarbeiter in einer Firma und diese sind wiederum ganz allgemein Personen. Es wäre also unnötig, all die Fähigkeiten und Eigenschaften dieser Objekte jedesmal neu zu beschreiben. Deshalb bildet man einen Stammbaum der Klassen, in dem die untergeordneten Klassen Eigenschaften und Fähigkeiten von ihren übergeordneten Klassen benutzen können. Dieser Baum bildet in objektorierten Programmiersprachen eine Klassenhierarchie. Werden Eigenschaften und Fähigkeiten aus dieser Klassenhierarchie benutzt, läuft ein Prozeß, den man Vererbung nennt. All diese Begriffe und Mechanismen werden im folgenden noch detaillierter erklärt.

Beziehungen zwischen Objekten führen zur Repräsentation von Wissen über die Realität in Form von semantischen Netzen[122]. In objektorientierten Systemen kann man drei Ausprägungen von Beziehungen identifizieren:

---

119 Fähigkeiten ermöglichen Verhalten.

120 Die hier verwendete Terminologie orientiert sich an den im Umfeld der Entwicklung von Smalltalk-gebräuchlichen Bezeichnungen. Es werden zusätzlich die englischen Bezeichnungen verwendet, weil die deutschen Übersetzungen in einigen Fällen unglücklich gewählt und irreführend sind.
Vgl. Goldberg, A., Robson, D. (Smalltalk, 1989), S. 5 ff.

121 Exemplare werden in der Umgangssprache objektorientierter Terminologie auch oft Instanzen genannt. Dies resultiert aus der ungenauen Übersetzung der englischen Bezeichnung instances.

122 Unseld, S. (Künstliche Intelligenz, 1990), S. 46f.

1. Die Klassenhierarchie führt zu einem Baum von ‘is a’-Beziehungen zwischen Klassen und übergeordneten Klassen.

2. Der Nachrichtenaustausch zwischen den Objekten legt ein implizites Netz von Abhängigkeiten in Form von Kommunikationsbeziehungen.

3. Sonstige Semantik kann durch die Erzeugung von beliebigen Verknüpfungen erzielt werden. So kann beispielsweise zur Verringerung der Komplexität von Systemen eine Hierarchisierung von Systemen und Subsystemen über ‘part of’-Beziehungen (... ist Teil von ...) erreicht werden.[123]

### Objektorientierte Analyse und Design[124]

Die soeben beschriebenen Ideen werden für die Durchführung von Projekten in zwei wesentliche Aufgabenkomplexe zerlegt: Die Verfahren zur Objektorientierte Analyse (OOA) und Objektorientiertes Design (OOD) beschreiben systematisch die Vorgehensweise, Ausdrucksmittel und Prinzipien für diese Aufgaben:

*Tabelle 15: Objektorientierte Analyse und Objektorientiertes Design*

| Aufgabenkomplex | Aufgabe |
|---|---|
| Objektorientierte Analyse (OOA) | Identifizierung der Objekte |
| | Festlegung der statischen und dynamischen Beziehungen zwischen diesen Objekten |
| | Festlegung struktureller Beziehungen |
| | Beschreibung der Eigenschaften und Fähigkeiten der Objekte |
| Objektorientiertes Design (OOD) | Umsetzung des aus der OOA hervorgebrachten Modells in eine konkrete Implementierung |
| | Nutzung der objektorientierten Softwaretechnologie |

Diese Reihenfolge der Argumentation und Vorgehensweise entstammt der Tradition der von *Berard* als „OOP Language People“[125] bezeichneten Autoren: Sie bauen OOA und OOD konsequent auf die durch jahrelange Erfahrung bewährten Grundprinzipien der Objektorientierung; so wie sie (von ihnen selbst) in den objektorientierten Programmiersprachen (OOP, vgl. 4.1.2) implementiert wurden. Vertreter sind beispielsweise *Beck* / *Cunningham*, *Wirfs-Brook* / *Wilkerson* oder *Rubin* / *Goldberg*. Als Gegenbewegung

123 Aspekte der Implementierung werden in den Abschnitten zur objektorientierten Programmierung und zu objektorientierten Datenbanken wieder aufgegriffen (siehe 4.1.2 und 4.1.3).

124 Vgl. zum folgenden: Hruschka, P. (Designprinzipien, 1994), S. 16 ff.

125 Berard, E. zitiert in: Thelen, B. (Ordnung, 1994), S. 80

identifiziert *Berard* die Gruppe der „Structured Crowd“[126], die systematisches Vorgehen - von OOA über OOD zur Implementierung - in den Vordergrund stellen. Die Autoren dieser Gruppe, wie *Shlaer* / *Mellor* oder *Coad* / *Yourdon*, stehen in der Tradition früherer Konzepte, die Datenstruktur-, Datenfluß- und Zustandsdiagramme als Ausgangspunkt ihrer Betrachtungen wählen. Die methodische Sauberkeit bei der Anwendung objektorientierter Konzepte läßt bei vielen dieser Ansätze sehr zu wünschen übrig. Die Folge ist, daß gerade dieser Ansatz in großen Projekten wegen der erschwerten Kommunikation mit dem Kunden versagen kann. Aus diesen beiden gegensätzlichen Positionen entwickelte sich ein lebhafter Methodenstreit, der sich in letzter Zeit durch die ‘Erkenntnis’ annähert, daß eigentlich die Aufgabe - das zu lösende Problem des Kunden - im Mittelpunkt der Bemühungen stehen sollte.[127]

Es ist genau dieser Aspekt, der im Abschnitt 4.4 aufgegriffen wird: Anforderungs-orientierte Planspielentwicklung mit objektorientierten Methoden.

Für die Planspielentwicklung soll daher der methodisch sauberere Weg der OOP Language-People eingeschlagen werden: In der Folge wird der objektorientierte Ansatz in seiner Reinform nach seinen Möglichkeiten und Grenzen für diese Aufgabe unter-sucht.

### Vorteile der Objektorientierung

Die Erfahrung in der Praxis hat gezeigt, daß mit der objektorientierten Methodik Auf-gaben von wesentlich höherer Komplexität bewältigt werden können, als mit der tradi-tionellen, prozeduralen (strukturierten) Vorgehensweise.

Warum kann die objektorientierte Methodik mehr leisten?

#### Semantische Lücke

> Mit Hilfe der strukturierten Programmierung versucht man, die Aufgabe nach Funktionen zu untergliedern, die auf vorher definierte Datenstrukturen (interne Speicherstrukturen und externe Ablage in Dateien und Datenbanken) zugreifen. Andere Funktionen nutzen die Kommunikationsschnittstelle von graphischen Benutzeroberflächen und Netzwerkkomponenten. Besonders bei den beiden letztgenannten Funktionen wird der Bruch in der Modellierung deutlich: Kommunikation, also Nachrichtenaustausch, wird durch Funktionsaufrufe und Datenmodellierung nachgebildet - ein Zusatzaufwand, der durch das objekt-orientierte Paradigma vermieden werden kann.

126 Berard, E. zitiert in: Thelen, B. (Ordnung, 1994), S. 80

127 Vgl. Thelen, B. (Ordnung, 1994), S. 79 ff.

### Keine künstlichen Schnittstellen

In der datenorientierten Denkweise der strukturierten Programmierung müssen künstlich Schnittstellen geschaffen werden, die das Zusammenarbeiten der verschiedenen Module ermöglichen. In objektorientierten Systemen sind diese Schnittstellen bereits natürlich durch die Nachrichtenschnittstelle vorgegeben.

### Bewältigung der Komplexität

Die Folge der größeren semantischen Lücke traditioneller Ansätze ist die ungenügende Unterstützung natürlicher Strategien zur Komplexitätsreduktion[128]: **Metamodelle** gröberer Granularität und deren Verfeinerungen in Submodellen können nicht direkt modelliert und in Form verständlicher Dokumentation von Struktur und Verhalten kommuniziert werden. Das für fundierte, und realitätsnahe Modellbildung notwendige Einbringen von Domänenfachwissen der Experten und die Validierung der Modellierung wird dadurch zusätzlich erschwert.

Objektorientierung hingegen bietet die Möglichkeit, Analyse, Design und Implementierung bis hin zur Schnittstelle mit dem Benutzer ohne Paradigmenwechsel durchzuführen - und damit zusätzliche, verfälschende Transformationen bei der Modellierung zu vermeiden.

## 4.1.2. Objektorientierte Programmierung

Die Konzepte und Ideen objektorientierter Programmierung gehen Hand in Hand mit OOA und OOD.

- Der folgende Abschnitt faßt deshalb zuerst die Grundzüge der objektorientierten Programmierung kurz zusammen.

- Wie die verschiedenen Programmiersprachen diese grundlegenden Konzepte interpretieren und implementieren, wird anschließend vorgestellt.[129]

Die erste Programmiersprache mit objektorientierten Elementen war Simula, eine Sprache, die in den Sechziger Jahren speziell für die Modellierung von Simulationsproblemen entwickelt wurde. [130]

---

128 Vgl. Freiburghaus, M. (Simulation betriebswirtschaftlicher Systeme, 1993), S. 54

129 BOOCH, einer der treibenden Kräfte in Sachen OOD, gibt in der Sektion „concepts“ seines Grundlagenwerkes „Object Oriented Design with Applications“ einen sehr guten Überblick über die Konzepte objektorientierter Systeme. Hier finden sich alle im folgenden aufgeführten Begriffe. Vgl. Booch, G. (OOD, 1991), S. 1ff.

130 Eine kurze Beschreibung der von *Nygaard* entwickelten Sprache Simula 67 findet sich z.B. in: Frauenstein, T., Pape U., Wagner O. (Sprachprinzipien, 1990), S. 80 ff.

Anfang der Siebziger Jahre wurde dann in den XEROX-Laboratorien mit Smalltalk die Muttersprache aller folgenden objektorientierten Sprachen entwickelt. Aus den dort entwickelten Konzepten bedienen sich beinahe alle folgenden objektorientierten Programmiersprachen mehr oder weniger intensiv.

### Merkmale objektorientierter Programmiersprachen

Im Laufe der Entwicklung objektorientierter Konzepte haben sich Merkmale der Objektorientierung herausgebildet:[131]

#### Kapselung

Unter Kapselung (engl. encapsulation) versteht man die Abschottung von Objekten nach außen. Objekte werden durch ihre Eigenschaften (engl. properties) und Fähigkeiten, implementiert in Methoden (engl. methods), beschrieben. Die Eigenschaften (=Daten) können in streng objektorientierter Sichtweise nur von den Objekt-eigenen Methoden und nicht direkt von außen manipuliert werden. Man spricht deshalb auch von information-hiding.

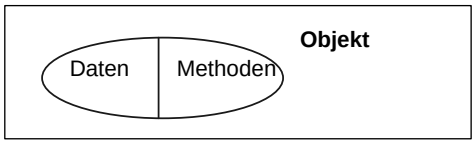



*Abbildung 4: Kapselung*

Kapselung vermeidet die bei der konventionellen Programmierung so gefürchteten Seiteneffekte, bei denen Funktionen Datenbereiche unberechtigt manipulieren. Solche Fehler sind meist besonders hartnäckig und oft für Qualitätsmängel von Software (Laufzeitfehler, mangelnde Stabilität, Systemabstürze) verantwortlich.

#### Nachrichtenschnittstelle (messaging)

Das Versenden von Nachrichten (engl. message) ist die einzige Möglichkeit mit Objekten in Kontakt zu treten ist. Eine Nachricht kann folgende Form haben:

<empfangendes Objekt> <Nachricht (Methodenaufruf)>  <Parameter>

131 Es hat sich mittlerweile eine gängige Terminologie zum Thema objektorientierte Programmierung etabliert. Da jedoch die meisten Ideen und Konzepte bereits für Smalltalk entwickelt und beschrieben wurden, sei nochmals auf Goldberg A., Robson, D. (Smalltalk, 1989) als Quelle verwiesen.

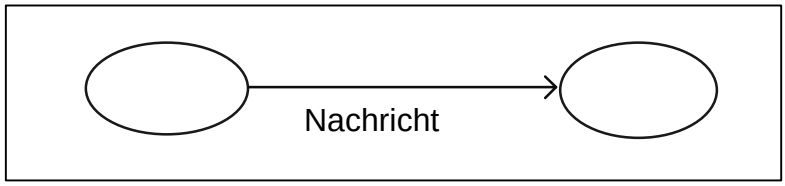



*Abbildung 5: Nachricht*

Das Objekt reagiert auf Nachrichten nur, wenn es eine entsprechende Methode kennt, die mit der <Nachricht> übereinstimmt. Optional schickt das empfangene Objekt nach Abarbeitung der Methode einen Rückgabewert an das <sendende Objekt> zurück.

Ein rein objektorientiert konzipiertes Programm besteht somit ausschließlich aus dem Nachrichtenaustausch zwischen Objekten. Ein Programm kann somit als Graph in Form eines Netzes aufgefaßt werden: Objekte sind die Knoten, Nachrichten werden entlang der Kanten gesendet.

### Klassenhierarchie

Wie bereits besprochen, werden gleichartige Objekte zu Klassen zusammengefaßt. Klassen legen den allgemeinen Bauplan (Eigenschaften und Fähigkeiten) für Objekte fest. So ist es nicht nötig, dies bei jedem Objekt noch einmal zu tun. Normalerweise sind die am Programmgeschehen beteiligten Objekte individuelle Exemplare (engl. instances) dieser Klassen. In Programmiersprachen, die über ein Metaklassenkonzept verfügen (also etwa Smalltalk), können auch Klassen wie ganz normale Objekte benutzt werden[132].

Da auch Klassen Gemeinsamkeiten haben, können sie in einer Hierarchie (engl. class hierarchy) angeordnet werden. Dieses Prinzip nennt man Ableitung von Unterklassen (engl. subclassing): Untergeordnete Klassen (engl. subclasses) sind Spezialisierungen[133] ihren übergeordneter Klassen (engl. superclasses). Klassen, die lediglich künstlich als allgemeine Oberklasse für verschiedene Unterklassen implementiert werden, heißen abstrakte Klassen (engl. abstract classes). Solche abstrakten Klassen finden sich immer oben in der Klassenhierarchie und werden niemals instanziert. Eine ausgereifte Sammlung von abstrakten und normalen Klas-

132 Solche Klassen verfügen über Klassenmethoden und Klassenvariablen. Es macht dann Sinn, Klassen wie normale Objekte zu benutzen, wenn zum Beispiel Verwaltungsaufgaben für alle Mitglieder (Exemplare) einer Klasse benötigt werden. Eine gängige Aufgabe ist es, die Exemplare einer Klasse zu hinterlegen und so im direkten Zugriff zu haben.

133 Das Prinzip der Spezialisierung von Superklassen zu Subklassen ist eines der mächtigsten Prinzipien der objektorientierten Methodik. Spezialisiert werden nicht nur die Eigenschaften, sondern oft auch die Methoden. Wird das Spezialisierungsprinzip konsequent durchgehalten, ist eine gute Klassenhierarchie möglich: Sie ist tief gestaffelt statt breit gefächert, redundanzarm und nutzt Polymorphie konsequent. Eine solche Klassenhierarchie kann leicht erweitert und gewartet werden.

sen nennt man „framework“ [134]. Frameworks werden von Softwareherstellern angeboten und bilden die wichtigste Grundlage für effiziente objektorientierte Softwareentwicklung. Im Abschnitt 5.1 wird ein solches Framework für Versicherungsplanspiele vorgestellt.

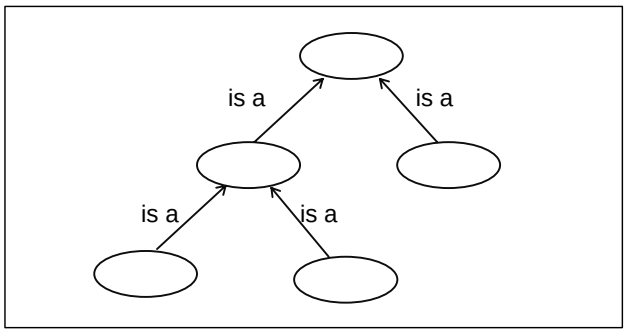



*Abbildung 6: Klassenhierarchie*

### Polymorphie

Da das Zusammenspiel der an einem Programm beteiligten Objekte ausschließlich über das Versenden von Nachrichten geschieht, macht es Sinn, sich an eine einheitliche Sprache zu halten. So werden dieselben Nachrichten zu verschiedenen Objekten gesendet und bewirken dort auch eine unterschiedliche Aktion. Dieses ‘Sprachspiel’ wird Polymorphie (engl polymorphism) genannt.

Ein Beispiel dafür ist die Nachricht <new>; jede Klasse reagiert auf diese Nachricht mit der Erzeugung einer Instanz, doch kann dieser Vorgang im Detail durchaus unterschiedlich aussehen - eben individuell an die Bedürfnisse angepaßt.

Der Vorteil einer solchen Vorgehensweise liegt klar auf der Hand: Für einen geübten Programmierer, der das Standardrepertoire der Nachrichten bereits kennt, ist das Erschließen der Funktionalität von neuen Klassen sehr viel einfacher. Die Lesbarkeit von Programmen wird durch Polymorphie deutlich erhöht.

### Vererbung

Klassen, die bestimmte Eigenschaften und Methoden nicht selbst implementiert haben, können auf die entsprechenden ihrer Superklassen zurückgreifen. Diesen Prozeß nennt man Vererbung (engl. inheritance). Vererbung ist ein elementarer Mechanismus zur Reduzierung der Redundanz innerhalb einer Klassenhierarchie. Erst durch Vererbung macht eine Klassenhierarchie überhaupt Sinn.

Einfache Klassenhierarchien sind als Baum ausgeprägt. Eine Klasse hat dann genau eine Superklasse. Vererbung dieser Art nennt man dann deshalb einfache Ver-

134 Zum Begriff des „frameworks“ vgl. Johnson, R. E., Foote, B. (Reusable Classes, 1988), S. 26

erbung (engl. single inheritance).
Hat eine Klasse mehrere Superklassen, so ist mehrfache Vererbung (engl. multiple inheritance) möglich. Für diesen Fall ist es notwendig, daß Algorithmen verwendet werden, die die Suchzeit nach der implementierenden Klasse optimieren. Mehrfachvererbung sollte aus Gründen der Übersichtlichkeit und Performance nur sehr dosiert eingesetzt werden.[135]

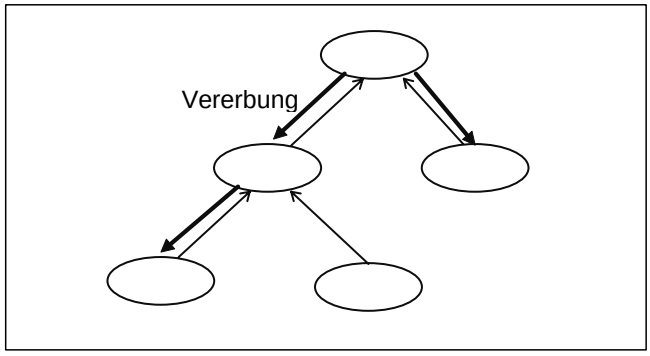



*Abbildung 7: Vererbung*

## Typengebundenheit

Klassische Programmiersprachen verlangen eine Typisierung der Variablen und damit der Parameter und Funktionen. Der Übersetzer (engl. compiler) reserviert dann den dem Typ (long, short, float, double, char etc.) entsprechenden Speicherplatz. Moderne Übersetzer überprüfen beim Übersetzen, ob die Parameterübergaben auch den Typ-Konventionen entsprechen. Ist dies nämlich nicht der Fall, sind Abstürze und Seiteneffekte nicht zu vermeiden.

Typengebundenheit (engl. strong typing) korrekt übersetzter Programme verbessert die Absturzsicherheit. In objektorientierten Entwicklungs- und Laufzeitumgebungen mit virtuellen Maschinen[136] ist aber Typengebundenheit nicht

135 Sieht man auf die neuere Entwicklung objektorientierter Systeme, wird über Vererbung kontrovers diskutiert: OLE 2.0, der 1993 vorgestellte Objektstandard von Microsoft, verzichtet ganz auf Vererbung, um auch im Netzwerk ein gutes Laufzeitverhalten bieten zu können und den Problemen bei Versionsänderungen aus dem Weg zu gehen.
Vgl. Puder, A. (Objekt, 1994), S. 54 ff.

136 Eine virtuelle Maschine ist der hardware- und betriebssystemabhängige Teil, auf den die objektorientierten Mechanismen aufsetzen. Bei streng objektorientierten Systemen können virtuelle Maschinen extrem klein und damit sehr gut portierbar gehalten werden. Wird sie hingegen in ein System wie Microsoft Windows® eingepflanzt, das bereits etliche Funktionen konventionell (= zu objektorientierten Prizipien inkompatibel) implementiert hat, so ist es eine durchaus schwierige Aufgabe, den Kompromiß zwischen Redundanz, Performance und Portierbarkeit zu finden. Man kann die unterschiedlichen Auffassungen sehr gut an Smalltalk-80 von ParcPlace und der Smalltalk/V-Implementierung von Digitalk beobachten.
Ein Beispiel für die Implementierung einer virtuellen Maschnine findet sich in: Goldberg, A., Robson, D. (Smalltalk, 1989), S. 567 ff.

unbedingt erforderlich - Programmabstürze werden von der Laufzeitumgebung abgefangen. Programmfehler haben somit so gut wie nie Folgen für die Betriebssicherheit der Applikation. Die Stabilität wird im wesentlichen von der Qualität der virtuellen Maschine bestimmt.

Der Preis der strengen Typengebundenheit ist, daß nicht einfach allgemeine Methoden einmal für alle Datentypen z.B. in einer abstrakten Klasse formuliert werden können. Die Möglichkeiten der Wiederverwendung von Methoden werden damit einschränkt.

### Späte Bindung

Späte Bindung (engl. late binding) bedeutet, daß vom Programm erst zur Laufzeit ermittelt wird, ob eine Nachricht von einem Objekt überhaupt verstanden werden kann. Deshalb nennt man späte Bindung auch manchmal dynamische Bindung. Typenfreie Programmierung erfordert späte Bindung.

Herkömmliche Compilersprachen hingegen legen beim Binden die (relativen) Einsprungadressen für die Funktionsaufrufe fest. Dieses Verfahren nennt man frühe Bindung (engl. early binding) oder auch statische Bindung (engl. static binding). Dies hat einen schnelleren Programmablauf zur Folge, da jedes Objekt von vorneherein weiß, welche Nachrichten es versteht und dann der zeitaufwendige und möglicherweise vergebliche dynamische Vererbungsprozeß wegfällt.

### Rein objektorientierte Programmiersprachen

Im folgenden sollen die wichtigsten objektorientierten Programmiersprachen kurz aufgeführt und bewertet werden. Dies geschieht vor allem, um die Wahl der geeigneten Programmiersprache für die Implementierung des in der vorliegenden Arbeit untersuchten Simulationsmodells nachvollziehbar zu machen.

Rein objektorientierte Programmiersprachen lassen die Konzepte der bisherigen Programmiersprachen der dritten Generation hinter sicht. Die konsequente Implementierung der objektorientierten Ideen soll deren Vorteile deutlich hervortreten lassen.

### Smalltalk

In XEROX-Laboratorien entwickelten seit Anfang der Siebziger Jahre *Goldberg*, *Ingalls* und *Kay* mit Smalltalk das Vorbild aller folgenden objektorientierten Programmiersprachen.[137]

137 Die aktuelle Sprachbeschreibung von Smalltalk liegt vor in: Goldberg, A., Robson, D. (Smalltalk, 1989)

Smalltalk verfolgt mit einer voll integrierten Entwicklungsumgebung kompromißlos den objektorientierten Ansatz. Das Konzept der virtuellen Maschine sorgt für Kompatibilität über alle implementierten Plattformen. Es wird ausschließlich typenfreie dynamische Bindung und einfache Vererbung verwendet. Automatische Speicherverwaltung (engl. garbage collection) und ausgereifte, mächtige Werkzeuge sorgen für produktive Entwicklung.
Smalltalk ist in zwei wesentlichen Implementierungen am Markt vertreten: Smalltalk-80, das ganz in der Tradition des unsprünglichen Smalltalk von ParcPlace auf allen gängigen Systemen implementiert ist, und Smalltalk/V von Digitalk.[138]
Als Hauptkritikpunkt an Smalltalk wird oft angeführt, die Performance der Systeme sei zu gering. Mit zunehmender Optimierung der virtuellen Maschine und der Erzeugung von maschinennäherem Zwischencode schmilzt der Unterschied zu C++ oft auf weniger als 10 oder 20%[139] zusammen. Lediglich für zeitkritische Echtzeitanwendungen ist Smalltalk noch nicht geeignet.

Für die Implementierung der objektorientierten Planspiel-Entwicklungsumgebung Object-VersPlan, wie sie im fünften Kapitel beschrieben wird, fiel die Wahl auf Smalltalk. Der Hauptgrund ist die konsequente Anwendung des objektorientierten Paradigmas, das vom Entwicklungswerkzeug bis hin zu den elementaren Klassen (Integer, Float ...) durchgehalten wird.

### EIFFEL

*Meyer* hat mit EIFFEL[140] ein vom Entwurf her überzeugendes objektorientiertes System geschaffen.

---

138 In Nordamerika gewinnt Smalltalk zunehmend Bedeutung für das Projektgeschäft. Bedingt durch die hervorragende Windows- und OS/2-Implementierung wird Smalltalk jetzt auch für größere Individual-Applikationen eingesetzt.
Eine Übersicht über die am Markt befindlichen Smalltalk-Systeme findet sich in:
Mittendorfer, J. (Vergleich, 1994), S. 45ff.

139 Der Performance-Vorsprung von C++ gegenüber Smalltalk könnte durch die Einführung von Standards wie dem RTTI weiter schmelzen. Dafür muß zusätzliche Funktionalität implementiert werden, die Smalltalk bereits standarmäßig in effizienter Form bietet.
Moderne virtuelle Maschinen bieten einen Übersetzer an, der den Smalltalk-Zwischencode zur Laufzeit in schnellen Maschinencode übersetzt. Eine Zwischenspeicherung (engl. caching) auf Methodenebene sorgt dafür, daß der dafür notwendige Mehraufwand durch Einsparung bei den Ausführungszeiten wieder wettgemacht wird. In einigen Fällen lassen sich Verbesserungen um bis zum Faktor 3 realisieren.
*Neatar* diskutiert die Problematik von Performancevergleichen zwischen C++ und Smalltalk und zitiert Projekte, bei denen von Smalltalk nach C++ portiert wurde.
Vgl. Naetar, F. (Smalltalk langsam?, 1994), S. 74 ff.

140 *Meyer* ist einer der treibenden Kräfte bei der Verbreitung der Ideen objektorientierter Softwareentwicklung. Er hat zum einen versucht, mit seiner Firma die von ihm entwickelte Sprache EIFFEL mit den nötigen Entwicklungswerkzeugen im UNIX-Umfeld zu vermarken und zum anderen auch durch zahlreiche Publikationen die dahinterstehenden Ideen verbreitet. Sein zum Standardwerk gewordenes Buch „Object-oriented Software-Construction“ ist als Einführung in die objektorientierten Grundsätze und als weiterführende Literatur gleichermaßen geeignet. Es gibt

Diese Sprache vereinigt alle Möglichkeiten rein objektorientierter Systeme mit den Vorteilen traditioneller Compilersprachen. Leider hat sich EIFFEL wegen der geringen Marktkraft seiner Hersteller nicht durchsetzen können und wird bisher lediglich in einigen Projekten vor allem im wissenschaftlich-technischen Bereich eingesetzt.

### Oberon

Die jüngste der hier vorgestellten objektorientierten Programmiersprachen ist die von *Wirth* entwickelte Sprache OBERON. Sie ist eine konsequente Weiterentwicklung der ebenfalls von Wirth entworfenen Sprachen Pascal und Modula-2. Mit OBERON versucht Wirth die Vorteile dieser Konzepte (Typisierung, Übersetzer, Kompaktheit, Module) mit den objektorientierten Ideen von Smalltalk (reine Objektorientierung, Portabilität durch Virtualisierung, mächtige Entwicklungsumgebung, **Model-View-Controller** (vgl. Abschnitt 4.2)) zu vereinen. Mit OBERON/F ist seit 1994 eine überzeugende Implementierung für eine Vielzahl von Plattformen verfügbar, die gerade im universitären Bereich immer mehr Anhänger findet.[141] Es hat sich jedoch noch kein nennenswerter Markt für Drittanbieter entwickelt. Nicht zuletzt deshalb kam OBERON für die Entwicklung von Object-VersPlan nicht in Betracht.

## Hybride Programmiersprachen und verwandte Konzepte

Hybride Programmiersprachen und verwandte Konzepte erweitern klassische Programmiersprachen um objektorientierte Elemente. Im folgenden seien nur die wichtigsten Sprachen und Spracherweiterungen aufgeführt.

### C++

*Stroustrup* (AT&T) hat C++ als Ergänzung zur Standardsprache C (von *Kernigan* und *Ritchie*) entwickelt.[142] Sie bietet gleichermaßen alle Möglichkeiten einer objektorientierten und einer systemnahen konventionellen Programmiersprache. C++ setzt im Stil einer klassischen prozeduralen Compilersprache auf Typenzwang und statische Bindung[143]. Deshalb können in C++ geschriebene Programme über

---

inzwischen eine deutsche Übersetzung, deren Titel mit "Objektorientierte Softwareentwicklung" allerdings etwas unglücklich gewählt wurde.
Siehe Meyer, B. (Software Construction, 1987)

141 Pountain, D. (Oberon/F, 1995), S. 227f.

142 Eine komplette Sprachbeschreibung findet sich in: Stroustup, B. (C++, 1992)

143 Mit dem neuen Standard von C++ und dem RTTI-Verfahren werden die Beschränkungen einer traditionell streng typisierenden Sprache durch die Verwendung von dynamischen Parametertabellen und Laufzeittypprüfung ein wenig aufgeweicht. Natürlich beeinträchtigt dies das Laufzeitverhalten. Die offizielle Verabschiedung und Veröffentlichung des neuen Standards ist für 1996 zu erwarten.

ein im Vergleich zu Smalltalk verbessertes Laufzeitverhalten bieten. Die Möglichkeit maschinennaher Programmierung und die weit verbreiteten Kenntnisse über C haben maßgeblich zur großen Akzeptanz von C++ beigetragen. Im Gegensatz zu Smalltalk bietet C++ Mehrfachvererbung.

Doch der Ansatz von C++ hat auch gravierende Nachteile: So bietet C++ standardmäßig keine automatische Speicherverwaltung. Dadurch entsteht im allgemeinen ein erheblicher Mehraufwand für die Implementierung objektorientierter Applikationen. Manche Untersuchungen sprechen von 30 bis 40 % gegenüber Smalltalk. Integrierte Entwicklungsumgebungen für C++ sind in der Regel im Vergleich mit Smalltalk-Systemen wenig mächtig und erfordern ein erheblich höheres Maß an Dokumentation. C++ eignet sich auch nicht sehr gut für den iterativen Software-Entwicklungsstil (engl. prototyping). Das Hauptproblem von C++ ist das Vorhandensein des vollen Sprachumfangs von C, der zu konventioneller prozeduraler Programmierung verführt. Dies behindert die Nutzung der Potentiale der Wiederverwendung. Trotz allem gilt C++ bislang als die objektorientierte Programmiersprache der Zukunft[144].

### Objective-C

Objective-C von *Cox*[145] spielt eigentlich nur auf dem Betriebssystem NeXT-Step[146] eine größere Rolle.
Die Sprache Objective-C ist mehr als C++ den objektorientierten Ideen gefolgt. Kern der Sprachbeschreibung ist die Erweiterung von C um (a) einen Datentyp für Objekte, (b) eine Syntax zur Beschreibung von Klassen und (c) eine syntaktische

---

Eine Vorab-Vorstellung der neuen Sprachelemente und die weitere Normierungsarbeit in den ANSI-Gremien findet sich in:
Hartinger, R. (Puzzle, 1994), S. 184 ff.
Hüskes, R. (Geduldsspiel, 1994), S. 180 ff.

144 Leider, denn man hat in Zeiten, in denen Performance nicht mehr die entscheidende Rolle spielt, eine gute Chance vertan, durch echte objektorientierte Umgebungen die Produktivität der Software-Entwicklung um ein noch größeres Stück zu steigern. C++ bietet Funktionalität,
die gute objektorientierte Grundsätze aufweichen: Es seien exemplarisch die Friends-Deklarationen (sie erschweren die Kapselung) und die Notwendigkeit virtueller Methoden (sie sind für Polymorphie notwendig) genannt. Auf der Konferenz „OOP '93“ in München wurden deshalb auch schon Stimmen laut, die an den Vorteilen der Objektorientierung ihre Zweifel angemeldet haben ... eigentlich klar, wenn sie alle C++ verwenden ...

145 Von *Cox* stammt die Bezeichnung wiederverwendbarer Objekte als "Software-ICs". Er legt dabei Parallelen von der Softwareentwicklung zur ingenieurmäßigen Entwicklung von Hardware und zeigt, wie der Einsatz objektorientierter Technologien unter konsequenter Ausnutzung der Wiederverwendungspotentiale das in der Software-Krise zum Ausdruck kommende Entwicklungsdefizit der Softwareentwicklung überwinden kann.
Siehe Cox, B. (OOP, 1986)

146 OpenStep, die portable Variante des NeXt-Step Betriebssystems und die auf dem Konzept der „Portable Distributed Objects“ basierenden Entwicklungsumgebung sind kurz beschrieben in:
Drespling, W., Lipps, P. (OpenStep, 1994), S. 26 ff.
Burger, E. (Communicate, 1994), S. 46 ff.

Struktur zum Versenden von Nachrichten. Eine Laufzeitunterstützung ermöglicht dynamische Bindung und macht damit typfreie Programmierung möglich.[147]

### ADA

ADA wurde im Auftrag des amerikanischen Verteidigungsministeriums als Sprache zur Verwendung bei großen Projekten entwickelt. Mit der Überarbeitung der Sprachnorm zu ADA-94 wird aus ADA eine vollwertige objektorientierte Programmiersprache: Jetzt verfügt sie über Möglichkeiten zur Kapselung, Klassenbildung und Spezialisierung, Vererbung und zum polymorphen Programmieren. Zusammen mit dem bewährten Modulkonzept der packages eignet sich ADA jetzt auch für die konsequent objektorientierte Entwicklung von großen und komplexen Softwaresystemen.[148]

### SOM und DSOM

IBM geht mit ihrem SOM (System Object Model) einen etwas anderen Weg: SOM erweitert jede Programmiersprache um objektorientierte Fähigkeiten wie Kapselung, Polymorphie und Vererbung. Als erste SOM-kompatible Klassenbibliothek sind die Objekte der Benutzeroberfläche Workplace-Shell von OS/2 2.x verfügbar. Auch bereits objektorientiert konzipierte Sprachen können von SOM profitieren. DSOM (Distributed System Object Model) ist die netzwerkfähige Variante für verteilte Objekte.

### CORBA

Vom Konzept her in eine ähnliche Richtung wie DSOM (und deshalb in Zukunft integriert) gehen die Standardisierungsversuche objektorientierter, verteilter Umgebungen[149], die von der Object Management Group (OMG[150]) vorangetrieben werden und als CORBA (Common Object Broker Request Architecture) bezeichnet wird. In Zukunft wird mit dem CORBA-Standard der OMG konsequent objektorientierte Architektur unabhängig von Programmiersprache und Betriebssystemplattform verfügbar.

---

147 Vgl. Droege, D. (Objektiv, 1994), S. 278 ff.

148 Vgl. Ebert, R. (Lady Ada, 1994), S. 146 ff.

149 Da die Standardisierungsbemühungen objektorientierter Architekturen intensiv diskutiert werden, gibt es zahlreiche Veröffentlichungen. *Beyer* gibt einen sehr guten, knappen Überblick.
Vgl. Beyer, T. (Objektbörse, 1993), S. 24 ff

150 Die OMG ist eine herstellerübergreifende Initiative zur Standardisierung von netzwerkweiten Objekt-Architekturen. Diese Architektur wird als Object Management Architecture (OMA) bezeichnet. Hauptbestandteil dieser Architektur ist die Common Object Request Broker-Architecture (CORBA). Sie legt fest, wie ein netzwerkweiter Object-Request-Broker (ORB) die Zugriffe auf Objekte regelt. COBRA-kompatible ORBs sind objektorientierte OODBMS, die über ein standardisiertes Protokoll, die IDL (Interface Definition Language), bearbeitet werden können.

### 4.1.3. Objektorientierte Datenbanken

Seit Anfang der achtziger Jahre verstärkt sich neben (oder auch ergänzend zu) den eingangs erläuterten objektorientierten Programmiersprachen auch die Diskussion über die Konzepte und Einsatzmöglichkeiten objektorientierter Datenbanken.

Da diese Konzepte für die Implementierung von Object-VersPlan zentrale Bedeutung haben, sollen kurz Prinzipien und Anwendungsmöglichkeiten vorgestellt und diskutiert werden.

#### Revolutionärer Ansatz[151]

Der als revolutionär beschriebene Ansatz objektorientierter Datenbanken basiert hauptsächlich auf den Prinzipien der objektorientierten Programmierung. Objektorientierte Datenbanken erweitern die Möglichkeiten objektorientierter Programmiersprachen:

#### Persistenz

Objekte, die während des Programmablaufs erzeugt werden, sind nach Beendigung des Programms normalerweise nicht mehr verfügbar. Eine objektorientierte Datenbank sorgt für die Speicherung und Wiederherstellung beliebiger Objekte.

Die Sicherstellung der Eindeutigkeit von Objekten (Objektidentität[152]) ist Aufgabe des Datenbanksystems und Voraussetzung für die Persistenz.

#### Datenbanksprache

Die Datenbanksprache enthält Elemente zur Manipulation und zur Abfrage: Die Sprachelemente zur Manipulation von objektorientierten Datenbanken ermöglichen die Definition, Erzeugung und Veränderung von Objekttypen und der auf sie möglichen Operationen. Die von der objektorientierten Programmierung bekannten Prinzipien der Vererbung und Instanzierung werden unterstützt. Die Abfragesprache ermöglicht das Wiederfinden von Objekten.
Ein Beispiel für eine vollständige Spracherweiterung ist die von *Kemper* und *Moerkotte* entwickelte Sprache GOM.

Gerade im Bereich der Sprachen für objektorientierte Datenbanken steht die von der Industrie akzeptierte Standardisierung noch aus: Erste Ansätze ergeben sich aus

---

151 *Kemper* und *Moerkotte* geben einen guten Überblick über die Basiskonzepte objektorientierter Datenbanksysteme. Sie gehen dabei hauptsächlich auf den revolutionären Ansatz objektorientierter Datenbanken ein.
Vgl. zum folgenden Kemper, A., Moerkotte, G. (Basiskonzepte, 1993), S. 69 ff.

152 Zur Sicherstellung der Objektidentität kann z.B. eine Kombination aus Schlüsselnummer (Object-ID), Version, Datenbankname und Datenbankort verwendet werden.

den bereits erwähnten Standardisierungsbemühungen der Object Management Group: Seit Ende 1993 liegt aufbauend auf den Standards OMA (Object Management Architecture) und CORBA (Common Object Request Broker Architecture) mit der ODMG-93 (Object Database Management Group) ein Standardisierungsvorschlag für objektorientierte Datenbanken zur Abstimmung mit den beteiligten Firmen vor. Ausgereifete, kommerzielle Produkte sind nicht vor Ende 1995 zu erwarten.[153]

Objektorientierte Datenbanken eignen sich besonders, wenn es gilt, komplexe Strukturen (z.B. Netz- oder Baumstrukturen) redundanzfrei abzubilden und zu speichern. Dazu bieten einige Datenbanken Konzepte zur Bildung von Verknüpfungen (engl. links) zwischen Objekten an. Die Abfragesprachen werden dazu um die entsprechende Semantik erweitert.[154]

### Transaktionskonzept

Eine schon bei konventionellen Datenbanksystemen unverzichtbare Fähigkeit ist die gesicherte Durchführung von umfangreichen, möglicherweise geschachtelten, Zugriffen auf die Datenbank. Wenn während Zugriffen, die Struktur und / oder Inhalt der Datenbank verändern, Fehler auftreten, kann mit Transaktionssicherung gewährleistet werden, daß der vor den Zugriffen bestandende Zustand wieder hergestellt werden kann.

Das Transaktionskonzept wird noch um eine (objektbasierte) Zugriffsrechteverwaltung erweitert, die das Design netzwerkfähiger Client-Server-Architekturen ermöglicht.

### Defizite konventioneller Datenbanken

Objektorientierte Datenbanken vermeiden viele Probleme konventioneller, relationaler Datenbanken[155]. *Kemper* und *Moerkotte* führen folgende Nachteile konventioneller

153 ODMG-93 (unter Federführung von *Catell*, SunSoft) definiert für objektorientierte Datenbanken eine Definitionssprache (Object Definition Language), eine Abfragesprache (Object Query Language), einen Baum von Objektklassen und legt die Implementierung der typischen Datenbankfunktionalität (Persistenz, Identität, Transaktionen, Zugriffsperren) fest. Darüberhinaus werden Schnittstellen zu den gängigen Programmiersprachen C++ und Smalltalk definiert.
Vgl. Behme, H. (Paradies, 1994), S. 118 ff.
Vgl. a. Gille, M. (ODMG 93, 1994), S. 70 ff.
Vgl. a. Atwood, Th. (Objekt-DBMS-Standard, 1994), S. 63 ff.

154 Bei der Implemetierung von Object-VersPlan wurde genau diese Spracherweiterung intensiv verwendet. Durch die Verwendung von links können die Zugriffe optimiert werden und die Speicherung der Objekte weitgehend redundanzfrei und performant geschehen.

155 Aus dem breiten Angebot der Veröffentlichungen zu konventionellen Datenbanken sei an dieser Stelle verwiesen auf: Schlageter, G., Stucky, W. (Datenbanksysteme, 1983)

Datenbankarchitektur auf und wie diese mit objektorientierten Datenbanken vermieden werden:

### Segmentierung

Objekte in der Realität müssen so zerlegt werden, daß sie ohne funktionale Abhängigkeit („normalisiert“) in durch Relationen verbundene Tabellen abgelegt („segmentiert“) werden können. Soll nun ein Objekt mit all seinen Eigenschaften geladen werden, müssen umfangreiche Verbundoperationen („joins“) durchgeführt werden. Dies kann zu erheblichen Effizienzeinbußen führen.[156]
In objektorientierten Datenbanken hingegen werden Objekte unsegmentiert als Ganzes gespeichert.

### Künstliche Schlüsselattribute

Damit die oben angeführten Relationen zwischen Tabellen erzeugt werden können, muß es möglich sein, Datensätze eindeutig zu identifizieren: Dazu legt man in der Regel künstliche Schlüsselattribute an. Viele Relationen führen zu einer erheblichen Menge systemweit zu vergebender künstlicher Schlüsselattribute, die zur eigentlichen Modellierung keinen Beitrag leisten.
Objektorientierte Datenbanken speichern Objekte mit Hilfe einer Objektidentität. Objektverknüpfungen über links benutzen diese Objektidentität zur Erzeugung beliebig komplexer Strukturen.

### Fehlendes Verhalten

Das anwendungsspezifische Verhalten der Objekte findet im relationalen Ansatz keinerlei Berücksichtigung: Verhalten muß von außen (außerhalb des Objektes) z.B. durch das Datenbanksystem realisiert werden.
Die in objektorientierten Datenbanken gespeicherten Objekte werden zusammen mit ihren Verhaltensmöglichkeiten (implementiert in den Methoden) abgelegt.

### Programmierschnittstelle

Die Datenmanipulationssprachen konventioneller Datenbanksysteme sind in ihrem Funktionsumfang stark beschränkt. Deshalb werden beispielsweise SQL-Abfragen in ein in der Sprache C geschriebenes Programm eingebettet. Zusätzliche Funktionen werden in C geschrieben. Diese Vorgehensweise vermischt die mengenorientierte SQL mit dem satzorientierten Arbeiten in C. Man spricht bei der

156 Praxisorientierter, guter Datenbankentwurf versucht, einen Kompromiß aus Redundanz (die das Laufzeitverhalten verbessern kann) und sauberem, problemnahem Entwurf zu finden. In der Regel wird deshalb auf vollständige Normalisierung verzichtet.

gleichzeitigen Anwendung mehrerer Paradigmen von „impedance mismatch“[157]. Objektorientierte Datenbanken, die in eine objektorientierte Programmiersprache eingebettet sind, vermeiden dies.

Zur Implementierung von Object-VersPlan 2.0 wurde ebenfalls eine objektorientierte Datenbank verwendet: „ODBMS“ des Braunschweiger Herstellers VC ist eine ganz im revolutionären Stil gehaltene objektorientierte Datenbank. Sie ist vollständig in die Programmierumgebung von Smalltalk eingebunden und implementiert die Konzepte Persistenz, Verknüpfungen, Versionsverwaltung, Transaktionsmanagement und Zugriffsrechteverwaltung[158].

### Evolutionärer Ansatz

Der evolutionäre Ansatz erweitert das relationale Datenmodell. Es wird versucht, die Vorteile des klassischen, relationalen Ansatzes und neuen Möglichkeiten objektorientierter Datenbanksysteme zu vereinen. Dadurch soll gerade im kommerziellen Bereich eine Koexistenz und Migration der beiden Ansätze erreicht werden.

Zwei Ansätze seien nur kurz erwähnt:

#### $NF^2$

Das relationale Datenmodell, das Tabellen in erster Normalform voraussetzt, wird dahingehend erweitert, daß Relationen auch als Attributwerte zulässig sind. Dieses „$NF^2$“ genannte Datenmodell ermöglicht hierarchische Abfragen und verbessert die Nachteile der oben beschriebenen Segmentierung.

#### SQL-3[159]

Die bisher recht verbreitete Datenbankabfragesprache SQL wird um der Objektorientierung nahestehende Elemente wie Sprachmittel für Datenbankprozeduren („stored procedures“), Ereignissteuerung („triggers“) und abstrakte Datentypen erweitert: Mit SQL-3 soll ein allgemein akzeptierter Standard zur Migration bestehender SQL-Lösungen geschaffen werden.

Bei der Modellierung komplexerer Strukturen gibt es allerdings noch Probleme mit der Integritätssicherung.

Für die Implemetierung von Object-VersPlan eignet sich der evolutionäre Ansatz nicht so gut, wie der eingangs beschriebene revolutionäre Ansatz:

---

157 Schmidt, J. W. (language constructs, 1977), S. 247

158 Vgl. Zerbe, K. (Plaudertasche, 1992), S. 110 ff.

159 Vgl. Weber, R. (SQL3, 1993), S. 95

– Evolutionäre objektorientierte Datenbanken lassen sich nicht nahtlos (also ohne Paradigmenwechsel) in ein konsequent objektorientiertes Entwicklungssystem integrieren.

– Die Modellierung sehr komplexer (manchmal auch zyklischer Zusammenhänge) wird ebenfalls nicht optimal unterstützt.

## 4.2. Möglichkeiten des objektorientierten Paradigmas für die Entwicklung von Planspielen

Die Idee zum Einsatz objektorientierter Technologien und Konzepte ist einerseits aus veränderten Anforderungen neuer Lehr-/Lern-Arrangements und andererseits aus den Defiziten konventioneller Ansätze zur Planspielentwicklung entstanden. Bietet das objektorientierte Paradigma Möglichkeiten zur deren Lösung?

Im folgenden sollen zuerst die Ansätze zur Implementierung von computerunterstützten Planspielen diskutiert und im Anschluß die Möglichkeiten des objektorientierten Paradigmas aufgezeigt werden.

### 4.2.1. Ansätze zur Entwicklung von Planspielen

Ansätze zur Entwicklung von Planspielen haben sich parallel zur der Weiterentwicklung der Methoden und Techniken der Softwareentwicklung[160] sowie der Simulation verändert.

Die Kriterien zur Einordnung der Ansätze heben ab auf die im dritten Kapitel herausgearbeiteten Anforderungen an die Planspielentwicklung:

*Tabelle 16: Kriterien zur Beurteilung von Ansätzen zur Planspielentwicklung*

| Kriterium | Frage |
|---|---|
| Modell | Folgt das Modell dem System-Ansatz? |
| | Besteht eine semantische Nähe zum Realitätsausschnitt? |
| | Kann die Komplexität des Modells anforderungsorientiert, variabel gehandhabt werden? |

160 Eine brilliante Analyse des Paradigmenwechsels im Software-Enginieering findet sich in: Quibeldey-Cirkel, K. (Paradigmenwechsel, 1994), S. 47 ff.

| | |
|---|---|
| Dokumentation | Wird das Modell in seinem Verhalten und seiner Struktur zielgruppengerecht kommuniziert? |
| Seminarsituation | Limitiert der Ansatz die Ausgestaltung der Anforderungen? |

Da Planspiele in ihrem Kern auf Simulationsmodelle und Verfahren zurückgreifen, wurde die Implementierung von Planspielen bislang auch im wesentlichen auf das Programmieren von **Simulationsmodellen** reduziert.

Es werden zuerst zwei aus der Simulationstechnik stammende Ansätze aufgeführt, die mit ihrem relativ hohem Grad an Allgemeingültigkeit auch für Planspiele verwendet werden können: Der System Dynamics-Ansatz und die Pertrinetz-Theorie.

### System Dynamics-basierte Lösungen

System Dynamics ist eine durch *Forrester* bekannt gewordene Methode zur Beschreibung komplexer, vernetzter Systeme durch Gleichungssysteme, die zur diskreten Simulation verwendet werden können.[161]

Modelle werden aus Grundbausteinen erstellt: Zentrale Systemelemente sind die Zustandsgrößen. Sie sind nicht durch andere Größen ausdrückbar und manifestieren mit ihren Veränderungen über die Zeit das Systemverhalten. Raten drücken diese Veränderung aus. Mathematisch gesprochen, sind die Veränderungen jeder Zustandsgröße in einer Differentialgleichung auszudrücken. Zu den Zustandsgrößen kommen als weitere Systembestandteile die Vorgabegrößen (also zum Beispiel Anfangswerte oder exogene Größen) und Zwischengrößen, die aus anderen Größen berechenbar sind.

Damit die dafür notwendigen, komplizierten mathematischen Formulierungen und Methoden vor dem Benutzer verborgen bleiben können, sind graphische Werkzeuge entstanden, mit denen Modellaufbau und Verhalten per Mausklick editiert werden kann. Als Beispiele seien - in aufsteigender Mächtigkeit - DYNAMIS[162], STELLA[163]und PowerSim[164] genannt.

Wie dargestellt wurde, gibt es für System Dynamics ausgereifte und starke Werkzeuge, die allerdings nur allgemein die Entwicklung des Modells und die Durchführung von Simulationen - nicht aber auf den Anwendungsfall Planspiel bezogen - unterstützen. Alle

161 System Dynamics entstand in den Sechziger Jahren am Massachusetts Institute of Technologiy (MIT). Weltweite Bekanntheit erlangte es durch die globalen Weltmodelle WORLD 2 und 3, die vom Club of Rome für die unter dem Titel „Die Grenzen des Wachstums“ bekannt gewordene Studie in Auftrag gegeben wurde.
Siehe Forrester, J. W. (Regelkreis, 1972); siehe auch Meadows, D. H. (Grenzen, 1972)
Eine Einführung zu System Dynamics findet sich in: Manhart, K. (Komplexität, 1994), S. 18 ff.

162 Vgl. Häuslein, A., Page B. (DYNAMIS, 1986), S. 329 ff.

163 Vgl. Simon, K.-H. (Realität spielen, 1992), S. 90 ff.

164 PowerSim wurde von der ASK der Europäische Hochschulsoftwarepreis 1994 verliehen.
Vgl. Schult, T. J. (Systemisches Denken, 1994), S. 64

hier betrachteten Werkzeuge bieten mächtige Möglichkeiten zur Dokumentation und Kommunikation des Modells; wohl auch deshalb, weil sie bei der Gestaltung der Benutzerschnittstelle dem objektorientierten Ansatz folgen.

System Dynamics liegt ein technisch-mechanistisches Weltbild zu Grunde - es können nur in Differentialgleichungen darstellbare Wirkungen und Zusammenhänge modelliert werden. Ebenso wird ein System als eine Vernetzung von Zustandsgrößen gesehen - Zustandsgrößen aber kommen in der Realität solitär aber gar nicht vor: Statt eines Kontostandes gibt es vielmehr ein Konto, das einen bestimmten Kontostand aufweist. Hier zeigt sich die semantische Lücke zwischen Realität und Modell.

Modelle, die mit diesen Werkzeugen erzeugt wurden, eignen sich prinzipiell gut dazu, Basis für klassische Unternehmensplanspiele zu sein. PowerSim bietet darüberhinaus durch seine offene Systemtechnik, seinen guten Möglichkeiten zur Modellstrukturierung (so zum Beispiel hierachisch gegliederte Modelle und nebenläufige Aktionen), den Werkzeugen zur Visualisierung und den ausgefeilten mathematischen Funktionen auch Möglichkeiten, Selbstlernsituationen zu unterstützen. Ist aber größere Interaktivität gefordert, so müssen diese Systeme wegen fehlender verteilter Architektur und dem Fehlen benutzerspezifischer Werkzeugen (für Teilenehmer und Spielleiter) versagen.

Trotz der Mächtigkeit der Werkzeuge bleiben die Probleme der Validität und Anwendbarkeit der Modelle bestehen:[165] Die hohe Allgemeinheit dieser Ansätze sieht nicht vor, domänenspezifisches Wissen über Systeme, Teilsysteme oder Elemente in einem Entwicklungssystem zu verankern. Experten, die über Domänenkompetenz verfügen, denen aber das detaillierte Verständnis für mathematische Zusammenhänge fehlt, kommen sehr schwer zu brauchbaren Ergebnissen. Damit kommt noch ein weiterer Kritikpunkt zur Sprache: Die Forderungen nach **Komplexität**svariation und Wiederverwendung bestehender Modelle werden nicht zuletzt durch die schwierige Kommunikation von Modellinternas erschwert.

### Petrinetz-basierte Lösungen

Mit der Theorie der Petrinetze liegt ein auf der Graphentheorie basierender Formalismus zur Beschreibung komplexer nebenläufiger Systeme vor. **Petrinetze** sind gerichtete Graphen, die aus aktiven und passiven Komponenten (Knoten) bestehen. Diese Knoten werden durch gerichtete Kanten verbunden. Auf dieser Grundlage haben sich verschiedene Netzklassen entwickelt, die sich zur Problemlösung unterschiedlicher Größenordnung eignen. Sie interpretieren die Funktionen von aktiven und passiven Elementen unterschiedlich:[166]

---

165 vgl. Simon, K.-H. (Realität spielen, 1992), S. 89

166 Im folgenden werden die vier Klassen von kausalen Petrinetzen in ihren Grundzügen vorgestellt. Für tiefergehende Erläuterungen sei verwiesen auf:
Hanisch, H.-M. (Petri-Netze, 1992), S. 19f.

1. Kanal/Instanzen-Netze (kurz: K/I-Netze) sind die einfachste Form von Petrinetzen: Kanäle sind die passiven Komponenten, Instanzen sind aktiv. Eine Komponente ist eindeutig einer dieser beiden Klassen zugeordnet. K/I-Netze eigenen sich zur Beschreibung der Struktur eines Systems, es entsteht ein statisches Netz von Prozeßeinheiten und Ressourcen, die Ressourcenrelationen unterhalten.

2. Bedingung/Ereignis-Systeme (kurz: B/E-Systeme) sind die wiederum einfachste der drei Arten dynamischer Petrinetze. Die Dynamikabbildung in Systemen erfolgt unter Verwendung von Bedingungen (passive Knoten) und Ereignissen (aktive Knoten); die Kanten entsprechen den Flußrelationen. Ob ein Ereignis eintritt („geschaltet wird"), hängt von seinen Konzessionen, also der Belegung der Vor- und Nachbedingungen durch Marken, ab.[167] Jede Markierung beschreibt einen Zustand des modellierten Systems. Zur Sicherstellung der Integrität wird Schleifenfreiheit gefordert. In B/E-Systemen sind zu jeder Zeit der aktuelle Zustand und die zu erwartenden Zustandsübergänge definiert. Die kausale Ordnung des Systems wird durch Erreichbarkeitsgraphen beschrieben. Aussagen über die zeitliche Ordnung sind mit klassischen Petrinetzen nicht möglich. Hierfür sind Erweiterungen notwendig, wie sie beispielsweise durch zeitbewertete Kanten eingeführt werden.[168] B/E-Systeme eignen sich für die Modellierung von Systemen geringer Größenordnung.

3. Stellen/Transitionen-Netze (kurz: S/T-Netze) erweitern das Markierungskonzept der B/E-Netze. In S/T-Netzen heißen die passiven Komponenten Stellen oder Plätze und die aktiven Komponenten Transitionen. Kanten kann eine Vielfachheit (ein natürliche Zahl) zugeordnet werden - sie gibt die Intensität an, mit der eine Transition ein Merkmal verändert. S/T-Netze können im Gegensatz zu B/E-Netzen auch Schleifen enthalten. Diese Erweiterungen erlauben die Modellierung von Systemen mittlerer Größenordnung und Komplexität.

4. Prädikat/Transitionen-Netze[169] (kurz: Pr/T-Netze) stellen die kompliziertesten der hier vorgestellten Petrinetze dar. In Pr/T-Netzen heißen die passiven Komponenten Prädikate und aktiven Komponenten Transitionen. Prädikate sind logische Formulierungen: Aussagen, die optional Variablen enthalten können. Jedem Prädikat ist eine Stellenzahl (natürliche Zahl) zugeordnet, die die Zahl der Variablen angibt. Prädikate mit positiver Stellenzahl tragen individuelle Marken, die nur mit Marken gleicher Stellenzahl belegt werden dürfen. Das Markenspiel wird über die Zuordnung von natürlichen Zahlen zu den Multimengen der an einem Prädikat beteiligten Objekte geregelt. Pr/T-Netze eignen sich für die Modellierung komplizierter Koppelungen in großen Systemen.

---

167 Dieses „Markenspiel" findet in allen Arten von Petrinetzen analog statt.

168 Vgl. Hanisch, H.-M. (Petri-Netze, 1992), S. 111ff.

169 Prädikat/Transitionen-Netze werden auch oft als gefärbte oder höhere Petrinetze bezeichnet.

Petrinetze haben sich vor allem in der Modellierung technischer Systeme (in der Informatik und in der Steuerungstechnik) etabliert[170] und erschließen sich zunehmend auch nichttechnische Anwendungsbereiche wie beispielsweise Systeme zur Vorgangsbearbeitung (engl. workflow-systems)[171]. Die Anwendbarkeit der Petrinetze wird durch Werkzeuge verbessert. Stellvertretend seien als Beispiele für Werkzeuge, die die Erstellung und die Durchführung von Petrinetz-basierten Simulationen unterstützen, FUN[172] und PSI-NET[173] genannt. Diese beiden Tools adressieren mit ihrem allgemeinen Ansatz eine breite Zielgruppe und können universell eingesetzt werden.

FUN

> FUN bietet die Werkzeuge für ein durchgängiges Konzept einer Software-Produktionsumgebung von der Analyse bis zur Programmgenerierung mit Petrinetzen. Die Modellierungsgrundlagen sind Funktionsnetze nach dem Schema der Kanal/ Instanzen-Netze. FUN ordnet den Instanzen Programmbausteine und den Kanälen Datentypen zu; Verknüpfungen von Instanzen und Kanälen repräsentieren Systembeziehungen.

PSI-NET

> PSI-NET ist ein graphisches Werkzeug zu Modellierung höherer Petrinetze. Diese „NET“ genannten Petrinetze erlauben neben qualitativer Dynamik durch zeitverzögertes Schalten auch die Abbildung zeitlicher Dynamik. Um die Modellierung sehr großer, komplexer Systeme zu unterstützen, kann durch Verfeinerung eine Modellhierarchie aufgebaut werden. Ein Laufzeitmodul und vielfältige Möglichkeiten der Auswertung und Darstellung unterstützen den Anwender.

Mit der Petrinetz-Theorie steht eine mächtige Methode zur Abbildung von komplexen Systemen in **Simulationsmodellen** zur Verfügung. Erweiterte Werkzeuge erlauben die Erstellung formal und syntaktisch korrekter Systeme ohne manuelle Programmierung und unterstützen die Durchführung der Simulation. Ohne fundierte Kenntnis des theoretischen Hintergrundes erscheint jedoch eine sinnvolle Anwendung nicht möglich. Dies ist der Grund, weshalb Petrinetz-basierte Lösungen vor allen in Bereichen Einzug gehalten haben, die ohnehin ein hohes Maß an mathematischem Formalismus benötigen. Beim Einsatz von Petrinetzen für die Entwicklung und Durchführung von Planspielen können daher die prinzipiellen Einschränkungen und Bedenken - wie sie bereits zum Thema

---

170 Vgl. Hanisch, H.-M. (Petri-Netze, 1992), S. 9f.

171 Vgl. Versteegen, G. (Hälse, 1995), S. 86 ff.
VERSTEEGEN stellt das Workflow-system LEU vor, das Petrinetze zur Modellierung, Simulation und Steuerung von Arbeitsabläufen benutzt.

172 Pape, U., Schoepf, V., Schwidder, K. (Petri Nets, 1994), S.1ff.

173 Vgl. Itter, F. (Integrierte Modellbildung, 1989), S. 90f.

System Dynamics-basierte Systeme aufgeführt wurden - analog in Betracht gezogen werden.

Die Vollständigkeits- und Integritätsbedingungen von Petrinetzen können als Basis für syntaktisch korrekte Modellierung dienen; sie machen jedoch keinerlei Aussage über die sachliche Richtigkeit und Anwendbarkeit Petrinetz-basierter Systeme. Ebensowenig stellen Petrinetze ein Modulkonzept zur Verfügung, das Wiederverwendung und anwendernahe Modellierung sicherstellt. Zwar führen Erweiterungen zu den klassischen Petrinetzklassen auch die zeitliche Dynamik ein; eine erweiterte Dynamik, die sich auf die Veränderung der Modellstruktur erstreckt, ist allerdings nicht vorgesehen.

Doch obige Ausführungen bedeuten nicht, daß Petrinetze keinen Beitrag zur Konstruktion von Planspiel-**Simulationsmodellen** leisten könnten. Grundsätzlich folgen sie dem Systemgedanken und lassen sich auch kompatibel zum objektorientierten Paradigma modellieren: Knoten und Kanten sind Systemobjekte und deren Beziehungen, Ereignisse werden durch Nachrichten übertragen und ausgelöst, Reihenfolgen und Bedingungen in komplexen Abläufen können durch das Markenspiel abgesichert werden. Dies zeigt sich beim Blick auf objektorientierte Systeme, wo beispielsweise die Prozeßverwaltung von **Smalltalk** Marken zur Prozeßkoordination verwendet.[174]

### Planspielgeneratoren und Individualentwicklungen

Planspielgeneratoren sind durch die Auslagerung allgemeiner Funktionalität (wie das Erstellen von Auswertungen bzw. von Eingabemasken für die Entscheidungen der Spieler oder der Marktsimulation) entstanden. Der Entwicklungsaufwand für anwendungsspezifische Planspiele soll so reduziert werden.[175]

Größere Anbieter von Planspielen benutzen zur Erstellung von kundenspezifischen Planspielen ebenfalls Modulbibliotheken, um den Aufwand für angepaßte Planspiele begrenzen zu können. Dieses Prinzip entspricht dem von Planspielgeneratoren, jedoch fehlen Werkzeuge zur Generierung von ablauffähigen Modellen. Hier ist die Benutzung von anwendungsneutralen Softwareentwicklungswerkzeugen notwendig.

Planspielgeneratoren und Modulbibliotheken leiden an den Beschränkungen konventioneller, strukturierter Softwareentwicklung: Die semantische Lücke, die zwischen Modell und Implementierung klafft, bringt unnötige, verzerrende Transformationen und Zusatzaufwand mit sich. Die Dokumentation (und damit die Kommunikation des Mo-

---

174 Im fünften Kapitel wird auf die Prozeßsteuerung in einer Planspielsimulation noch einmal eingegangen.

175 Planspielgeneratoren spielen keine große Rolle am Markt: Die Entwicklungsaufwendungen für allgemein anwendbare Werkzeuge sind hoch und der Markt für entsprechende Softwarewerkzeuge ist noch klein. Ein Beispiel ist der Planspielgenerator P1, der 1989 auf den Markt kam. Anbieter, die bereits über Wissen in diesem Bereich verfügen, tendieren dazu, entweder fertige, anwendungsspezifische Planspielprogramme und ganze Planspielseminare anzubieten oder als Entwickler für individuelle Planspiele aufzutreten.

dells) wird stark erschwert, da sie in einer Art Einbahnstraße erstellt wird: Interaktion und Dynamik können nicht berücksichtigt werden. Planspiele, die mit Generatoren und Modulbibliotheken erzeugt wurden, eignen sich daher trotzdem nur für relativ standardisierte Anwendungsfälle, die in klassischer Seminarsituation oder als Fernplanspiel abgehalten werden.

### Beispiele für neuere Ansätze

In den Markt für Planspiele und Simulationen ist mit fortschreitender Verbreitung von Hard- und Software Bewegung gekommen. Innovative Lösungen, die Anregungen für weitergehende Entwicklungen bieten, sind beispielsweise CAPS und CABS.

#### CAPS[176]

> CAPS ist ein in erster Linie für die Untersuchung von Produktionsprozessen entwickeltes Simulationssystem, das sich an der Systemtheorie orientiert: Mit Hilfe eines graphischen Editors ist es möglich, Systemelemente - versehen mit Eingangsvariablen, Zustandsvariablen, Ausgangsvariablen und Funktionen, die diese Werte beeinflussen - miteinander zu verbinden: Es entsteht ein ereignisgesteuertes Prozeßmodell. Über die Verbindungen können auch Werte weitergereicht, jedem Simulationsschritt kann eine Zeit zugeordnet werden. Das so entstandene Prozeßmodell kann direkt in der Entwicklungsumgebung simuliert werden. Darüberhinaus stehen Funktionen zur Überprüfung der Konsistenz des Modells und zur Visualisierung von Systemgrößen zur Verfügung.[177]
>
> Mit seiner systemorientierten Konzeption und der freundlichen Benutzerschnittstelle zeigt CAPS den Weg, den simulationsorientierte Werkzeuge gehen können. Für Unternehmensplanspiele bietet CAPS lediglich einen Rahmen, jegliche anwendungsspezifische (=planspielspezifische) Funktionen fehlen jedoch. Zwar wird auf der Ebene der Systemelemente Ereignisorientierung und Objektorientierung verfolgt, ein Blick in das Innenleben der Systemelemente zeigt allerdings Konzepte prozeduraler Programmierung.

176 Vgl. Mikro Partner (CAPS, 1991), S. 1.1 ff.; vgl. a. Simon, K.-H. (Realität spielen, 1992), S. 93

177 CAPS ähnelt in seinem Design den zeitbewerteten Petrinetzen.

#### CABS[178]

CABS ist eine interaktive Unternehmenssimulation, die konkurrierende Unternehmen der Automobilindustrie im europäischen Markt simuliert, die die Funktionsbereiche Absatz, Marketing, Produktion, Logistik, Produktentwicklung und die Stabsfunktionen des Rechnungswesens beinhaltet. Die Nutzung von CABS als Selbstlernumgebung ist möglich, die Konkurrenzunternehmen werden vom Rechner geführt. Der Konkurrenzmodus erlaubt die Durchführung eines klassischen Planspielseminars, es besteht jedoch die gravierende Beschränkung auf lediglich zwei Unternehmen. Aus Sicht des Spielleiters und der Teilnehmer sind - wie bei allen 'fertigen' Unternehmensplanspielen - Struktur und Zusammenhänge fix modelliert, ebenso die verfügbaren Auswertungen und Visualisierungen.

Der folgende Abschnitt widmet sich dem objektorientierten Ansatz und diskutiert, welche Möglichkeiten sich aus dessen konsequenter Anwendung für die Entwicklung von Unternehmensplanspielen ergeben können.

### 4.2.2. Modellbildung und Implementierung bei objektorientierter Planspielentwicklung

Wie der vorangegangene Abschnitt gezeigt hat, ist die semantische Lücke zwischen Problem und Implementierung eines der gravierendsten Probleme konventioneller Ansätze zur Planspielentwicklung.

Sind die im Abschnitt 4.1 vorgestellte objektorientierte Methodik und deren Möglichkeiten zur Implementierung in der Lage, diese Probleme und Beschränkungen überwinden zu helfen?

Ein Blick auf die objektorientierte Bewältigung der beiden Abbildungsprobleme der Planspielentwicklung, Modellierung und Implementierung, gibt hier Aufschluß:

#### Objektorientierte Modellierung von Planspielen

Der objektorientierte Ansatz spielt bei der Modellierung seine Stärke aus: Die semantische Lücke zwischen realer Welt und objektorientiertem Modell wird sehr klein gehalten.

---

178 Vgl. Virtual Management (Alles, 1994), S. 4 ff.
CABS wurde mit Visual Basic (VB) implementiert: VB ist eine Mischung aus den Ideen objektorientierter Programmierung für das Design der Benutzeroberfläche und der prozeduralen Programmiersprache BASIC für die Implementierung aller weitergehenden Funktionen. Es fehlen jedoch so wichtige Eigenschaften wie Vererbung und dynamische Bindung. Konsequenz dieses uneinheitlichen Sprachkonzepts ist die unzureichende Modularisierung der Programme: So kann auch CABS keine ausreichenden Möglichkeiten zur individuellen Anpassung bieten.

Folgende Tabelle zeigt die relevanten Transformationen von der realen Welt in die Welt des objektorientierten Modells:

*Tabelle 17: Transformation der realen Welt im objektorientierten Ansatz*

| reale Welt | objektorientiertes Modell |
|---|---|
| Systembestandteile | Objekte (Exemplare von Objekten) |
| Eigenschaften, Fähigkeiten, Verhalten | Attribute, Methoden |
| Beständigkeit | Persistenz in objektorientierter Datenbank |
| Spieler und Spielleiter | Benutzer |
| Kommunikation | Kommunikation |
| Interaktion, Sprache | Nachrichtenaustausch über Protokoll |
| Beschreibung von Struktur und Verhalten | Dynamische Sichten ('views') auf Objekte und Gruppen von Objekten |
| Beziehungen | Datenbankverknüpfungen ('links') |
| statisch | Hierarchie, sonstige Semantik |
| dynamisch | Aktionen über Nachrichtenfluß |
| Aktivitäten und Prozesse | Methoden und Prozesse |
| Bedingungen, Auslöser | Semaphore |
| Zeit | Spiel-Perioden |

Es wird sehr deutlich sichtbar, wie gering Verluste sein können, wenn Tatbestände der realen Welt in einem objektorientierten Modell abgebildet werden.

### Systembestandteile und Objekte

Grundlegende Transformation bei der objektorientierten Modellbildung ist die Abbildung der Systembestandteile in Objekte. Die Erkenntnis, daß sich viele ähnliche Elemente als gleiche Typen klassifizieren lassen, führt in vielen objektorientierten Systemen zur Bildung von Klassenhierarchien. Am Systemgeschehen nehmen aber strenggenommen nur die Exemplare dieser Klassen teil. Es ist gleichbedeutend, ob es sich bei den Systembestandteilen um Elemente oder wiederum um Systeme selbst handelt; beide werden als Objekte aufgefaßt. Systemelemente haben im objektorientierten Verständnis (in direkter Abbildung der Realität) Eigenschaften und Fähigkeiten, die ihr Verhalten bedingen.

Um die Lebensdauer von Objekten über die Programmlaufzeit hinauszuschieben, werden objektorientierte Datenbanken (vgl. 4.1.3) verwendet.

### Kommunikation

Ein elementar wichtiger Aspekt der Realität ist die Kommunikation, sehr allgemein als Austausch von Informationen zwischen den Systembestandteilen verstanden. Objektorientierte Methodik setzt hier mit einer nahen Analogie auf - einem sehr weit ausgelegtem Kommunikationsbegriff. Neben der Übertragung von Information, wie es in der realen Welt durch Sprache oder Zeichen geschieht, versteht Objektorientierung auch das Anstossen von Aktionen aller Art als Kommunikation. Der Sprachumfang des Empfängers wird als Protokoll abgebildet.[179]

### Beziehungen und Verknüpfungen

Folgt man dem Systemverständnis, so sind die Beziehungen zwischen Elementen konstitutiv. Im objektorientierten Modell können Beziehungen sehr leicht und flexibel nachgebildet werden. Jedes klassenbasierte, objektorientierte System (also zum Beispiel Smalltalk oder C++) kennt als elementare Beziehung die ‘is_a’-Verknüpfung: Eine spezialisierte Unterklasse ist gleichzeitig vom Typ ihrer Superklasse, Exemplare sind vom Typ ihrer Klasse. Die Klassenhierarchie per se eignet nicht für die Abbildung von Systemen, sie ist vielmehr eine Meta-Ebene zur Beschreibung eines Systems. Die Abbildung weiterer Beziehungen erfolgt im Rückgriff auf die überaus flexible Möglichkeit der ‘links’: Mit ihnen lassen sich Beziehungen aller Art beschreiben. (In wieweit die Implementierung den gesteckten Anforderungen genügen kann, wird in Verlauf noch diskutiert.) Implizit werden Beziehungen über die Nachrichtenschnittstelle hersgestellt.

### Aktivitäten und Prozesse - Bedingungen und Auslöser

Schon nahe der Implementierungstechnik ist die Abbildung von Aktivitäten und Prozessen in objektorientierten Systemen. Hier gibt es verschiedene Ansätze, aus denen beispielhaft die Lösung Smalltalks herausgegriffen sei: Damit auf einer im Prinzip seriell verarbeitenden Rechnerarchitektur parallele Aktivitäten und Aktivitätsketten bearbeitet werden können, wurde mit dem Prozessmodell eine sehr einfache und praktikable Lösung gefunden: Kontrolliert von einer zentralen Instanz, die dem Prozessor nahesteht, werden Aktivitäten in voneinander getrennten Prozessen in Zeitscheiben je nach Priorität

179 An dieser Stelle soll keine eingehende Diskussion über den Begriff der Kommunikation geführt werden. Der hier angesprochene Kommunikationsbegriff entspricht im wesentlichen der Mathematischen Kommunikationstheorie von *Shannon* und *Weaver*.
Siehe Shannon, C. E., Weaver, W. (Communication, 1949)
In den Wirtschafts- und Sozialwissenschaften wurden diese technisch orientierten Ansätze, die als Kommunikationspartner Menschen und Maschinen gleichermaßen betrachten, bereits früh kritisiert. Psychologische, sozialpsychologische und soziologische Ansätze legen den Schwerpunkt auf die Kommunikation zwischen Menschen und betonen deshalb die Verhaltenskomponente (Einstellungen, Werte und Normen, Selektionsprozesse, Beziehungsaspekte, Reflexivität und Reziprozität).
Vgl. Merten, K. (Kommunikation, 1977), S. 12 ff.

abgearbeitet. Zur Synchronisation einander bedingender oder sich auslösender Aktivitäten werden Semaphore verwendet.[180]

### Zeit

Erst durch die Zeit entstehen dynamische Systeme. Im Modell für ein Unternehmensplanspiel wird die Zeit durch die Spielperiode abgebildet. Dabei kann je nach Anforderung von weiten Zeitsprüngen im Jahrestakt bis zur quasi-kontinuierlichen Periodenabständen variiert werden.

Im Gegensatz zur Realität, ist die Zeit im Modell eine globale Größe - eine zulässige Vereinfachung, die im täglichen Leben (... in dem die spezielle Relativitätstheorie keine Rolle spielt ...) ebenfalls gemacht wird.

### Vom Modell zur Implementierung

Wie bereits im vorangegangenen Abschnitt besprochen wurde, kann dieses Modell mit Hilfe objektorientierter Programmiersprachen umgesetzt werden. Die Stärke liegt jetzt in der Durchgängigkeit der Methoden: Die Paradigmen zur Modellierung und zur Implementierung sind identisch - die semantische Lücke ist geschlossen. Konsequenterweise ergibt sich hieraus, daß die Schwierigkeiten bei der Umsetzung gering gehalten werden können.

Trotzdem sind kritische Aspekte zu bedenken:

- Probleme treten dort auf, wo die symbolorientierte Logik Beschränkungen auferlegt, also etwa bei der Abbildung von assoziativem Verhalten oder der Formulierung von Heuristiken. Durch die Kombination mit wissensbasierten Systemen, die sich objektoriert sehr gut abbilden lassen, und der Anwendung der Fuzzy-Logik Technik[181] können diese Defizite gemildert werden.[182]

- Da die Implementierung zu anwendbaren Systemen führen soll, muß auf einschränkende Nebenbedingungen wie zum Beispiel die Hardware-Ressourcen oder verfügbare Entwicklungsplattformen Rücksicht genommen werden. Der notwendige Kompromiß ‘verwässert’ jedoch die Sauberkeit der Implementierung.

---

180 Details zur Verwendung von Prozessen und Semaphoren in Smalltalk-Systemen finden sich in: Goldberg, A., Robson, D. (Smalltalk, 1989), S. 249 ff.

181 Grundlagen zur Fuzzy-Logik und deren Anwendungen finden sich in: Tilly, T. (Fuzzy-Logik, 1991), S. 13ff.

182 Vgl. Unseld, S. (Künstliche Intelligenz), S. 44 ff.

## Objektorientiert zum Planspiel

Damit die aufgeführten Vorteile in der Praxis auch genutzt werden können, muß die Planspielentwicklung von den Softwareentwicklern und -anbietern unterstützt werden.

Basis ist eine objektorientierte Planspiel-Entwicklungsumgebung (engl. developers workbench). Sie hilft dem Konstrukteur von Planspielen mit einem Satz mächtiger, anwendungsspezifischer Werkzeuge, sich auf die Abbildung der Anforderungen des Anwendungsfalles (wie im dritten Kapitel beschrieben) zu konzentrieren. Die Zeit für Entwurf, Entwicklung und Test kann niedrig gehalten werden. Erst mit spezialisierten Werkzeugen ist produktive Entwicklung möglich. Durch die hohe Anwenderorientierung verschiebt sich die Zielgruppe solcher Werkzeuge hin zum Domänenfachmann - die sonst üblichen Kommunikationsprobleme zwischen Entwickler, Programmierer und Experten für Pädagogik und Anwendungsgebiet des Planspiels werden entschärft. Im fünften Kapitel werden Beispiele für Werkzeuge vorgestellt.

Zwingende Voraussetzung für die Anwendung dieser Werkzeuge ist ein Pool vorgefertigter, ausgetesteter Objekte. In objektorientierten Umgebungen wird dies eine Klassen- oder Objektbibliothek in Form eines Frameworks sein:

### Allgemeine Simulationsobjekte

Kontext-unabhängige Anforderungen werden durch allgemeine Simulationsobjekte abgedeckt. Diese Objekte haben einen sehr hohen Wiederverwendungsgrad, da sie in den verschiedensten Anwendungsfällen (Kontexten) zum Einsatz kommen können. Die Eigenschaften und Fähigkeiten allgemeiner Simulationsobjekte legen die Einsatzmöglichkeiten für die verschiedenen Seminarsituationen (vgl. 3.2.1) fest; sie stehen in einer objektorientierten Klassenhierarchie nahe der Wurzel.

Damit sich ein Markt bildet, ist ein breit akzeptierter Standard für Objekte notwendig. Zum heutigen Zeitpunkt ist gerade die Festlegung des Standards für verteilte Objekte durch die OMG erfolgt, was jedoch nur die unbedingt notwendige, systemtechnische Voraussetzung bedeutet. Standards für den Anwendungsfall Unternehmensplanspiel sind von Normierungsgremien nicht zu erwarten: Hier werden sich bestenfalls de-facto-Standards einzelner Anbieter etablieren können. Die aus Sicht der Anwender gewünschte breite Kompatibilität von Werkzeugen und Objekten unterschiedlicher Hersteller erscheint unrealistisch. Für große Unternehmen, Institutionen der öffentlichen Verwaltung, großen Bildungseinrichtungen oder Planspielanbieter lohnt die Einhaltung eigener Standards: Objektorientierung bietet dafür bessere Möglichkeiten als jede andere Entwurfs- und Implementierungstechnik.

### Spezialisierte Simulationsobjekte

Mit den allgemeinen Simulationsobjekten allein läßt sich nur dann ein anforderungsgerechtes Planspiel konstruieren, wenn die Granularität sehr fein und damit die Allgemeingültigkeit sehr groß ist. Dann schmelzen aber die Produktivitätsvorteile wegen des geringen Wiederverwendungsgrads und des hohen eigenen Entwicklungs- und Testaufwand dahin.

Aus Sicht eines Anwenders ist es deshalb wünschenswert, wenn vorgefertigte, spezialisierte Objekte bereits eng an den Kontext des Anwendungsfalls angepaßt sind. Je nach Häufigkeit und Bedeutung der Anwendung können die Objekte einer vorgefertigten ‘Fachbibliothek für Simulationsobjekte’ entnommen werden oder aber es bedarf für den speziellen Einsatzzweck einer Neuentwicklung.

Folgt man konsequent den Prinzipien wiederverwendungsorientierter Objektentwicklung[183], so wird die Neu- oder Weiterentwicklung noch stärker spezialisierter Objekte effizient und wirtschaftlich sein.

Der folgende Abschnitt greift diese grundsätzlichen Überlegungen auf und konkretisiert sie zu Lösungsansätzen für die spezifischen Probleme bei der Planspielentwicklung.

Im fünften Kapitel wird am Beispiel einer konkreten Implementierung gezeigt, wie eine allgemeine Objektbibliothek für Unternehmensplanspiele und eine daraus abgeleitete **Bibliothek** für den Anwendungsfall Versicherungsplanspiel aussehen könnten.

## 4.3. Bewältigung der Problemdimensionen der Planspielentwicklung

Um ein komplexes Problem bewältigen zu können, wird es vor dem Hintergrund der vorher festgelegten Anforderungen (vgl. 3) in ‘handlichere’ Teile zerlegt, die einer Lösung leichter zugänglich sind. Dies soll aber nicht bedeuten, die Grundidee des Systemansatzes zu begraben - das Systemmodell als Ergebnis dieser Bemühungen steht als Ganzes immer im Blickpunkt.

Die nun folgenden Lösungsansätze greifen auch nicht genau einen Aspekt oder ein Problem auf, sie bieten vielmehr - ganz im Sinne des objektorientierten Ansatzes -

183 Drei Beispiele für Prinzipien wiederverwendungsorientierter Entwicklung seien aufgeführt:
1. Entwicklung abstrakter Klassen, die weit oben in der Klassenhierarchie aufgehängt werden.
2. Ableitung spezialisierter Objekte in durch die Bildung von Unterklassen (engl. subclassing).
3. Kombination bestehender Objekte zu neuen Objekten, ohne in die Klassenhierarchie eingreifen zu müssen.

durchgängige Konzepte, die die im vorangegangenen Kapitel aufgestellten Forderungen an Modellbildung und Implementierung von Planspielen erfüllen helfen.

1. Die Zerlegung des Problems mit Hilfe des Modell-View-Controller-Ansatzes legt die Basis für eine strikte Trennung des eigentlichen Planspiel-Modells von seiner Dokumentation und Kommunikation mit Hilfe spezialisierter Werkzeuge.

2. Die getrennte Betrachtung der Statik (der **Aufbauorganisation**) und der Dynamik (der Ablauforganisation) reduziert die **Kompliziertheit** des Modells.

3. Die Frage nach Mikro- oder Makro-Modellierung beeinflußt maßgeblich den Charakter und die Anwendung des Simulationsmodells.

4. Sind diese grundsätzlichen Lösungen zur Modellbildung und Implementierung gefunden, können Möglichkeiten zur **Komplexitätsvariation** herausgearbeitet werden. Mit dieser Aufgabe ist das Problem der Realisierung wiederverwendbarer, auch bei Veränderung der Modellstruktur, aufwärtskompatibler Objekte verbunden.

### 4.3.1. Model-View-Controller - das grundlegende Gliederungsprinzip

Wie jedes Computerprogramm, so hat auch ein Programm für ein Unternehmensplanspiel drei grundlegende Aufgaben zu erfüllen:

- Es modelliert ein Problem der Realität und versucht, eine Lösung anzubieten.

- Es präsentiert sich dem Benutzer auf dem Bildschirm oder einem anderen Ausgabegerät.

- Es interagiert mit dem Benutzer (dem Spielleiter, Konstrukteur oder Teilnehmer), indem es auf Eingaben und Befehle reagiert.

#### Subject-View-Controller (S-V-C)

*Rumbaugh* zeigt, wie diese grundlegende Erkenntnis unabhängig von der Programmiersprache nutzbringend in objektorientiert konzipierte interaktive Applikationen eingesetzt werden kann.[184] Er identifiziert, bezeichnet und beschreibt drei Ebenen:

- Subject (im folgenden mit ‘Subjekt’ übersetzt) beinhaltet die semantische Information, die direkt zum Komplex des abzubildenden Problems gehört. Diese Informationen und Aktionen (Berechnungen) finden sich auch in der Realität wieder -

184 Vgl. hierzu und zum folgenden: Rumbaugh, J. (MVC, 1994), S. 49 ff.

sie stammen aus dem Domänenfachwissen eines Experten. Es ist dabei für die Subjekte völlig unerheblich, wie sie sich dem Benutzer präsentieren oder mit ihm interagieren.

– Der View[185] bietet Möglichkeiten der Aufbereitung von Information über Subjekte für den Benutzer. Dies kann in allen für den Benutzer wahrnehmbaren und vom Computer produzierbaren Formen geschehen - meist also visuell oder auch als Ton. *Rumbaugh* faßt dies kurz in „Ein View ist die Projektion seines Subjekts“[186]. Es werden in der Regel verschiedene Views für die verschiedenen Informationen eines Subjekts benötigt.

– Die Ebene der Controller[187] kümmert sich um den interaktiven Aspekt eines Problems. Sie isoliert den aktuellen Steuerungskontext (engl. control state) der Applikation von dem eigentlichen Problem und seiner Präsentation.

Durch diese Trennung (insbesondere vom View-Objekt) ergeben sich optimale Möglichkeiten für die Wiederverwendung und die Redundazarmut objektorientierter Systeme, da gerade Controller- und View-Funktionalitäten häufig in sehr ähnlicher Form verwendet werden.

Die folgende Abbildung zeigt die Schichten und Abhängigkeiten im Grundgerüst dieses Subject-View-Controller genannten Ansatzes:

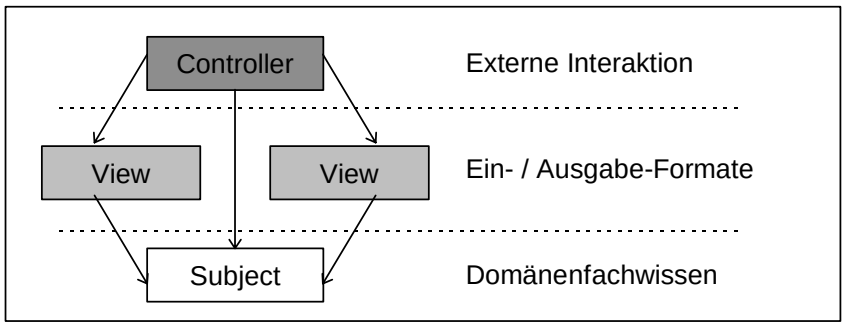



*Abbildung 8: Schichtenmodell der Subjekt-View-Controller Architektur[188]*

185 Der Begriff View soll im folgenden englisch verwendet werden, da die deutsche Übersetzung zusätzliche Ungenauigkeiten und Mißverständnisse über diesen feststehenden Begriff erzeugen würde.

186 Rumbaugh, J. (MVC, 1994), S. 50

187 Controller ist in der objektorientierten Terminologie ein feststehender Begriff und soll deshalb nicht übersetzt werden.

188 Rumbaugh, J. (MVC, 1994), S. 49

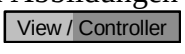
Zwischen Subjekt und View existiert eine statische Abbildungsvorschrift, die den Subjekten die jeweils passenden Präsentationsmöglichkeiten zuweist. Hinzu kommt eine dynamische Komponente; die Konsistenz von Subjekt und View muß gewahrt werden. Sinnvollerweise übernimmt dabei jeder View die Verantwortung für die Aktualität seiner Präsentation, indem er Informationen über sein Subjekt besitzt. Von den Zuständen der anderen Views weiß ein View-Objekt hingegen nichts.

### Smalltalks' Model-View-Controller (M-V-C) [189]

**Smalltalk**-Programmierumgebungen wissen von dieser Trennung in die Ebenen Subject-View-Controller und implementieren sie als **Model-View-Controller**-Architektur [190].

In klassischen Smalltalk-Implementierungen wie Smalltalk-80[191] findet man Model-Objekte und Paare von View/Controller-Objekten:

– Nur etwas, das sich dem Benutzer präsentiert, kann auch von ihm manipuliert werden - deshalb werden View und Controller zur Programmlaufzeit fest verbunden.

  Anmerkung: In den Abbildungen der vorliegenden Arbeit werden View-Controller-Paare immer durch 

dargestellt.

– Jedem dieser Paare ist genau ein Model (im folgenden mit Modell übersetzt[192]) zugeordnet. Ein Modell hingegen kann beliebig viele View/Controller-Paare benutzen. Deshalb wird ein Abhängigkeitsmechanismus (engl. dependence) implementiert, der alle View/Controller zu einem Modell bekannt macht. Wenn er

---

189 *Hüskes* gibt einen kurzen, sehr gut verständlichen Abriß zu M-V-C und dessen Implementierung. Vgl. Hüskes, R., (Musketiere, 1993), S. 174 ff.

190 Die aktuelle Auflage der Sprachbeschreibung und der Konzepte von Smalltalk findet sich in: Goldberg, A., Robson, D. (Smalltalk, 1989)
Die vom *Rumbaugh* gewählte Bezeichnung Subjekt-View-Controller erscheint sauberer, da der Begriff des Modells jetzt ohne die Gefahr der Verwechslung in seiner Standardbedeutung (die abstrakte Beschreibung eines Systems) benutzt werden kann. Er übernimmt aber trotzdem die allgemein gebräuchliche und kurze Bezeichnung View, obwohl es es sich, allgemein gesprochen, um die Ebene der Präsentation handelt.
Da Object-VersPlan in Smalltalk implementiert wurde, soll im folgenden die gebräuchliche Bezeichnung Model-View-Controller verwendet werden.

191 Anmerkung: Die für Object-VersPlan verwendete Smalltalk-Implementierung von Digitalk setzt M-V-C nicht konsequent um. Dies ist in der nahen Anlehnung an die Möglichkeiten der graphischen Benutzeroberflächen begründet. In der Implementierung von Object-VersPlan selbst wird M-V-C hingegen durchgängig verwendet.

192 Um einer Begriffsverwirrung vorzubeugen: Die Bezeichnung Modell bringt Probleme mit der Eindeutigkeit. In den vorangegangenen Erläuterungen zum Thema Modellbildung (vgl. 2.2) wurde mit Modell immer ein Abbild eines relevanten Realitätsausschnittes bezeichnet. Im M-V-C Ansatz bezeichnet Modell wiederum denjenigen Ausschnitt aus diesem Modell, der sich um die logische Ebene kümmert. Deshalb sollen diese Objekte mit Modell-Objekt bezeichnet werden. Die Gesamtheit der Abbildung heißt weiterhin Modell.

nach Änderungen eines Modells aktiviert wird, können die View/Controller die Präsentation aktualisieren.

Mit dieser Art der Arbeitsteilung erreicht man in einem ersten großen Schritt für die Implementierung bereits die Reduktion der Kompliziertheit auf das für die Modellierung des eigentlichen Problems erforderliche Mindestmaß: Man kann sich primär den Modell-Objekten widmen, ohne für spätere Erstellung interaktiver und am Benutzer orientierter Applikationen Zugeständnisse machen zu müssen. Diese aus Sicht der Modellierung eher sekundären Probleme können dank M-V-C zu einem späteren Zeitpunkt angegangen werden.

Zahlreiche Projekte der Entwicklung mittlerer bis großer Applikationen haben gezeigt, daß M-V-C eine solide, ausgereifte Basis für redundanzarme und zugleich flexible Implementierung legt. Mit M-V-C können die Vorteile objektorientierter Softwareentwicklung, so die Möglichkeiten zur Wiederverwendung, besonders gut genutzt werden.

### Möglichkeiten für Planspiele

Gerade für Planspielentwicklung und -durchführung kann dieser Ansatz wertvolle Beiträge leisten: M-V-C ist die Basis für zielgruppenorientierte Kommunikation mit dem einem Planspiel zugrundeliegenden Simulationsmodell.

Beispiele sollen dies verdeutlichen:

Betrachtet man die Ebene des Modells, so könnte dies in einem Versicherungsplanspiel zum Beispiel ein Bestand von versicherten Risiken sein. Für globalere Betrachtungen kann die Repräsentation eines Kollektivs und die Ausweisung von Durchschnittswerten (beispielsweise der Schäden und der Versicherungssummen) ausreichend sein, während in Seminaren mit dem Schwerpunkt Bestandsmanagement ein detaillierter Ausweis der einzelnen Verträge notwendig sein wird: Hierfür muß das passende Modell in einer Objektbibliothek zur Verfügung gestellt werden. Es kann von einem bestehenden Modell abgeleitet sein oder durch die Kombination bereits bestehender Modelle erzeugt werden. Der Ballast der Repräsentation dieses Modells und die Beschäftigung mit Aspekten der Benutzerschnittstelle spielt in diesem Zusammenhang keine Rolle - nur die Abbildung des Problems selbst, also dem relevanten Ausschnitt der Realität, ist von Belang.

Die Interaktion mit diesem Modell wird erst durch ein View-Controller-Paar möglich. Die Repräsentation (View) etwa der Schadenentwicklung des oben angeführten Versicherungsbestandes könnte zum Beispiel durch eine Tabelle, eine Grafik oder auch nur durch eine Art Ampel[193] geschehen. Die Detailliertheit der vermittelten Informationen kann auf diese Weise dosiert werden. Sollen Informationen durch den Benutzer verertet

193 Eine Art Ampel könnte durch die Farben grün, gelb oder rot signalisieren, ob sich ein zu beobachtender Wert innerhalb oder außerhalb der gewünschten Bandbreite bewegt.

werden, muß dem View ein passender Controller zugeordnet werden: In Tabellen blättert und editiert man, Grafiken zeigen sich in verschiedenen Typen, Ausschnitten und Ansichten, und für eine Ampel müssen beispielsweise die Schwellenwerte der Farbabstufungen eingegeben werden können.

Durch die vielfältigen Kombinationen von View-Controller-Paaren und deren flexible Zuordnung zu Modell-Objekten erhält man ein hohes Maß an Flexibilität, um die Bedürfnisse des jeweiligen Benutzers zu befriedigen: Ein Spielleiter fordert wesentlich tiefergehende Möglichkeiten der Kommunikation mit dem Modell als dies ein Spieler tut, ein fortgeschrittener Spieler kann mit detaillierteren Informationen umgehen, als dies ein Anfänger kann, in einem dynamischen Planspiel und einer simulationsbasierten Selbstlernumgebung werden Werkzeuge (=View/Controller) benötigt, die ein höheres Maß an Interaktivität bieten, als dies in einem klassischen Planspielseminar der Fall sein wird.

Die Konsequenz für die Planspielentwicklung ist, daß eine flexible, schrittweise anpaßbare **Bibliothek** von Model-,View- und Controller-Objekten zu Verfügung stehen muß. Entsprechend kombiniert, kann sie helfen, die Bedürfnisse des Anwendungsfalles soweit wie möglich zu erfüllen.

### 4.3.2. Statik und Dynamik - zwei Sichtweisen der Modellierung

Bei der Vorstellung der grundlegenden Ideen des Systemverständnisses (vgl. 2.1) wurde bereits das Komplexitätsproblem angesprochen: Durch die Vielzahl der Systembestandteile des Modells und ihrer intensiven Vernetzung entsteht der Bedarf nach Methoden zur Bewältigung der Komplexität durch Verringerung der Kompliziertheit: Warum reduziert man also nicht die Verschiedenheit der Subsysteme, Elemente und Relationen durch Klassifizierung auf ein zu bewältigendes Maß?

Ein Seitenblick auf die Modellierungstechniken zur objektorientierten Softwareentwicklung gibt hierzu einen Hinweis: *Rumbaugh* schlägt in seiner Object Modelling Technique (OMT) die Bildung von drei Modellen vor:[194]

- Das Objektmodell enthält alle Gegenstände der Applikation.
- Das dynamische Modell bildet das interaktive Verhalten des Systems und dessen Beschränkungen ab.
- Das Funktionsmodell beschreibt Operationen, Berechnungen und Transformationen.

194 Vgl. Kavanagh, D. (OMT, 1994), S. 59 ff.
*Kavanagh* stellt die erweiterte Anwendung der bereits 1991 von *Rumbaugh* entwickelten Methode OMT vor und erweitert sie für den Einsatz in der Praxis um neuere Ansätze.

Für die objektorientierte Entwicklung eines Planspiel-**Simulationsmodells** kann eine analoge Aufteilung in Teilmodelle gewählt werden:

### Objektmodell

Das Objektmodell eines Planspiels kann auf zwei Modelle reduziert werden: Die Klassenhierarchie und die **Aufbauorganisation**. Es ist wichtig, den semantischen Unterschied zwischen diesen beiden Modellen zu würdigen:

In einer Klassenhierarchie werden allgemeine Beschreibungen von Objekten abgelegt:

- Eine baumartige Hierarchie entsteht durch die Spezialisierungsrelation ‘is_subclass’ (manchmal auch mit: ‘is_a’ bezeichnet)

- Klassenhierarchien sind grundlegender Bestandteil objektorientierter Systeme. Eine Programmiersprache wie **Smalltalk** bringt von sich aus schon alle notwendigen Mittel zur Beschreibung mit.

Das Modell der Aufbauorganisation enthält alle Elemente des Simulationsmodells. Es bietet eine Momentaufnahme der Struktur - also eine Beschreibung der Statik des Modells.

- Aufbauorganisationen entstehen durch die Verbindung von Exemplaren der Klassenhierarchie mit der Semantik ‘part of’ (... ist Teil von ...).

- Wird das hierachische Systemkonzept (vgl. 2.1) auf Aufbauorganisationen angewendet, führt dies zu einer Zerlegung in Modelle und Submodelle: Im Fall eines Unternehmensplanspiels ist diese Hierarchisierung geradezu natürlich: Märkte werden in Teilmärkte zerlegt, Unternehmen haben eine Aufbauorganisation, die nach verschiedenen Kriterien (Sparten, Funktionen ...) in Teilhierarchien aufgeteilt werden kann.

### Dynamik-Modell

Die qualitative Dynamik des Modells wird durch die Interaktionen der Systemelemente beschrieben: Im objektorientierten Ansatz also durch ein Geflecht von Nachrichtenaustausch, Bedingungen und Reihenfolgen. Parallele Aktionen unterstützt ein Prozeßmodell.[195]

Aufbauend auf den in jedem objektorientierten System verfügbaren Nachrichtenmechanismus wird für jede Klasse eines Systemelements das Protokoll zur Verwendung in der

195 Näheres zum Prozeßmodell findet sich in den Abschnitten 5.1.2 und 5.2.3.

Dynamik-Modellierung definiert: Man kann dies als eine Untermenge der mit Mitteln der zugrundeliegenden Programmiersprache implementierten Methoden verstehen.

Auch Dynamik-Modelle gewinnen durch die Bildung von Hierarchien an Übersichtlichkeit: Es entsteht ein rekursives Modell von Abläufen, die, für sich allein betrachtet, wesentlich weniger komplex sind als das Gesamtmodell.

Quantitativ dynamische Modelle können beispielsweise durch Zeitbewertung der Nachrichtenverbindungen realisiert werden - ein Konzept, das bereits im Rahmen der Theorie höherer Petrinetze (vgl. 4.2.1) angewendet wurde.

#### Funktionsmodell

Ein Funktionsmodell, wie es OMT in der Tradition von strukturierter Analyse und Design[196] vorsieht, wird nicht zusätzlich benötigt: Funktionen finden sich in den Methoden der Objekte implementiert. Funktionsmodellierung ist nicht kompatibel mit den in der Objektorientierung wesentlichen Prinzipen der Kapselung und der Polymorphie.

Fazit: Zur Abbildung komplexer Modelle reichen durch die Anwendung des objektorientierten Ansatzes schon wenige Hilfsmittel zur Beschreibung aus: Verbindungen mit der Semantik ‘part_of’ und ‘is_a’, hierarchische Gliederung in Modell und Submodell, und die konsequente Verwendung der Prinzipien objektorientierter Programmierung.

### 4.3.3. Mikro oder Makro - die entscheidende Frage für die Modellbildung

Im Abschnitt über Simulation (vgl. 2.2) wurde bei der Diskussion der Dynamik bereits ein Aspekt angesprochen, der von entscheidender Bedeutung für die Modellierung einer Planspiel-Simulation ist: Das Problem der Aggregation - prägnanter als Mikro/Makro-Problem bezeichnet.

#### Problemstellung

Um eine Vorstellung von dieser Problematik zu bekommen, kann ein Blick in die Volkswirtschaftslehre weiterhelfen: Sie unterscheidet prinzipiell eine mikro- und eine makroökonomische Betrachtungsebene: Während im ersten Fall die Untersuchung des Verhaltens individueller, wirtschaftlicher Entscheidungseinheiten (Haushalte oder Unternehmen) im Vordergrund steht, werden im zweiten Fall die Beziehungen von Aggregaten betrachtet.[197]

---

196 Techniken zur strukturierten Analyse und Design sind die durch CASE-Werkzeuge am meisten verbreiteten Entwicklungsansätze. Sie wurden für die prozeduralen, datenorientierten Programmierkonzepte entwickelt.

197 Vgl. hierzu und zum folgenden: Buteweg, J. (Systemtheorie, 1988), S. 51 ff.

Basis der mikroökonomischen Betrachtungsweise sind Überlegungen zum methodologischen Individualismus von *Popper, Watkins, Mises, von Hayek* und anderen. Sie vertreten (stark verkürzt) die Ansicht, daß alles Wissen über soziale Phänomene nur aus dem Wissen über individuelles Verhalten abgeleitet werden kann.

Will man durch Aggregation von der Mikroebene zu makroökonomischen Aussagen kommen, setzt man sich den Schwierigkeiten der Aggregationsbildung aus: Qualitative Unterschiede und Unstetigkeiten gehen in den Glättungen des Aggregats unter, Rückkoppelungen auf der Marko-Ebene müssen nicht zwangsläufig ein Äquivalent in der Mikro-Ebene haben. Darüberhinaus gibt es für etliche makroökonomische Probleme keine mikroökonomische Fundierung (z.B. Konjunktur- und Geldtheorie).

Als Ausweg führt *Buteweg* die Argumentation an, daß allen makroökonomischen Betrachtungen implizit ein (unbekanntes) mikroökonomisches Modell zugrunde liegt. Soll die ceteris paribus-Klausel als Grundlage weiterhin gültig bleiben, bedeutet dies, daß sich das mikroökonomische Modell nicht allzu schnell ändert - und damit der makroökonomischen Betrachtung die Grundlage entzieht. Diese Argumentation wird noch deutlicher, wenn man der Aspekt der Strukturkausalität mit einbezieht: In manchen Bereichen ist es eher möglich, Regelmäßigkeiten auf der Makro-Ebene zu erkennen als auf der Mikro-Ebene. Während auf der Mikro-Ebene scheinbar ‘Chaos’ herrscht, können makroökonomische Gesetzmäßigkeiten beobachtet werden - das Mikro-System determiniert das Makro-System.[198]

198 Vgl. Buteweg, J. (Systemtheorie, 1988), S. 53 f.
Diese Mikro-Fundierung negiert die Eigengesetzlichkeit emergenter Phänomene.
*Luhmann* verneint in seiner „Theorie sozialer Systeme“ die Erklärung von Makro-Systemen durch die ihnen zugrundeliegenden Mikro-Systeme. Er bezeichnet den Mikro/Makro-Zusammenhang in Anlehnung an *Maturana* als „strukturelle Koppelung“ und erklärt aus Sicht des Makro-Systems das Mikro-System zur Umwelt, das es beeinflußt, aber nicht erklärt. (*Luhmann* verwendet allerdings eine völlig andere Begriffswelt, deren Vorstellung hier zu weit führen würde. Das Beispiel *Luhmanns* sieht das Gehirn und seine Neuronen als Mikro-System und das Bewußtsein als Makro-System.)
Vgl. Kneer, G., Nassehi, A. (Theorie sozialer Systeme, 1993), S. 62 f.

*Hannsmann* zeigt am Beispiel des Absatzbereichs die Bandbreite der Modellierung an vier sich an der Betrachtungsebene orientierenden Modelltypen auf:[199]

*Tabelle 18: Modelltypologie von Absatzmodellen*

| Globales Makromodell | Mikroanalytisches Modell | Partielles Makromodell | Mikroverhaltensmodell |
|---|---|---|---|
| Gesamtumsatz eines Unternehmens als Funktion hochaggregierter absatzpolitischer Variablen | Gesamtumsatz zurückgeführt auf die Beiträge einzelner Kunden - die selbst wieder abhängig sind von den Umwelteinflüssen und der absatzpolitischen Variablen | Disaggregation des Umsatzes nach Regionen, Produkten etc. mit individuellen Reaktionsfunktionen | Umsätze aus dem Kaufverhalten einzelner Kunden erklärt. Marketingmaßnahmen wirken auf dieses Verhalten. |
| geeignet für strategische Rahmenentscheidungen | geeignet, wenn aus der Gestaltung der Variablen strategische Stoßrichtungen abgeleitet werden sollen | geeignet für Entscheidungen auf der zweiten oder dritten Managementebene | geeignet, wenn bis ins Detail formulierte Marketingmaßnahmen abgeleitet werden sollen. |

Die Tabelle zeigt die Modelle in aufsteigender **Komplexität** von links nach rechts.

- Globale Makromodelle erfordern den geringsten Modellierungsaufwand. Sie können für Planspiele verwendet werden, deren Lernziele die Marktaspekte nur global betreffen: Dies sind beispielsweise Planspiele mit dem inhaltlichen Schwerpunkt Bilanzierung oder Finanzierung oder solche, die wegen ihrer Zielgruppe nur geringe Komplexität benötigen.

- Mikroanalytische Modelle setzen zwar auf den Beiträgen einzelner Kunden zum Gesamtumsatz auf, sie fordern jedoch keine spezifischen Informationen über das individuelle Verhalten einzelner Kunden oder Kundengruppen. Ein Beispiel mag dies verdeutlichen:

  $$U_t = K_t * A_t * D_t$$

  Der Umsatz einer Periode t ist eine Funktion der Anzahl der Käufer im Markt ($K_t$), dem Anteil, die sich für das Unternehmen entscheiden ($A_t$) und dem

199 Vgl. hierzu und zum folgenden: Hannsmann, F. (Quantitative Betriebswirtschaftslehre, 1985), S. 113 ff.

Durchschnittsumsatz mit einem Kunden ($D_t$). Da die Erklärungsvariablen den Umwelteinflüssen und den absatzpolitischen Gestaltungsvariablen ausgesetzt sind, erhöht sich die Komplexität der Modellierung. Für Planspiele ist diese Art der Modellierung insofern interessant, als sie den Teilnehmern ein bessere Entscheidungsgrundlage vermittelt, mit deren Hilfe sie ihre Marketingprogramme den Bedürfnissen der Kunden anpassen können.

- Partielle Makromodelle erlauben, die Komplexität der Modellierung gegenüber globalen Makromodellen durch Disaggregation - die Zerlegung in Teilmodelle mit unterschiedlichen Reaktionsfunktionen - schrittweise zu steigern. Es ist zu beachten, daß die Größe der Partialmodelle eine gewisse Mindestgröße nicht unterschreitet - sonst bewegt man sich auf einer Aggregationsebene, die nur für mikroökonomische Betrachtung geeignet ist. Für Planspiele ist das Bilden von Partialmodellen eine gute Möglichkeit, schrittweise Komplexitätssteigerungen und Anpassungen an reale Gegebenheiten vorzunehmen.
  Wie die Globalmodelle, so können auch Partialmodelle mit mikroanalytischen Modellen kombiniert werden.

- Mikroverhaltensmodelle übertreffen mit ihrer möglichen Komplexität und den Anforderungen an die benötigten Primärdaten zu Modelleinstellung die zuerst genannten Modelltypen. Es werden Informationen über das Verhalten von Kunden und Kundengruppen benötigt. Planspiele, die mit Hilfe von Mikroverhaltensmodellen beschrieben werden, stellen höchste Anforderungen an Technik und Teilnehmer - Mikroverhaltensmodelle können praktikabel in (abgeschlossenen) Teilmodellen verwendet werden, wenn es um die Abbildung von „schneller Mikrodynamik“[200] geht.[201]

Während Makromodelle und mikroanalytische Modelle mit qualitativer Dynamik und komparativer Statik gut repräsentiert sind und mit auch quantitativer Dynamik kombiniert werden können, bietet es sich an, Mikroverhaltensmodelle als quantitative dynamische Modelle zu implementieren. Für Unternehmensplanspiele kommt diese Art der Modellierung nur in Frage, wenn ein dynamisches Planspielseminar oder eine Selbstlernumgebung vorliegen. Qualitativ dynamische Modelle sind hingegen für alle Seminarsituationen einsetzbar.

Ein Bereich, an dem sich die **Mikro/Makro-Problematik** ebenfalls gut zeigt, ist die Modellierung von Organisationen wie beispielsweise Planspielunternehmen: Prinzipiell können die am Beispiel des Absatzbereichs vorgestellten Modelltypen hier angewendet werden. Wegen der im Vergleich zum Absatzmodell kleineren Kollektive (Gruppen von

200 Rapoport, A. (Systemtheorie, 1988), S. 67

201 Ein Beispiel aus dem Absatzbereich wäre hier die Modellierung der Außendienstanstrengungen durch eine Gruppe individuell ausgebildeter Außendienstmitarbeiter.

Mitarbeitern sind in der Regel kleiner als Kundengruppen) steht die Modellierung von Organisationen der Mikro-Modellierung näher.

### 4.3.4. Komplexität des Modells - Möglichkeiten zur Variation

Die im dritten Kapitel geführte Diskussion über anforderungsorientierte Planspielentwicklung hat als eine der grundlegenden Forderungen insbesondere diejenige nach variabler **Komplexität** des Modells postuliert. Im folgenden sollen deshalb Möglichkeiten und Probleme der **Komplexitätsvariation** in der objektorientierten Planspielentwicklung erörtert werden

#### Möglichkeiten zur Komplexitätsvariation

Die vorangegangenen Abschnitte über M-V-C, Statik und Dynamik und zum **Mikro/-Makro-Problem** der Modellierung bieten die notwendigen Ansatzpunkte zur Beschäftigung mit dem Problem der Komplexitätsvariation.

1. Ausgerichtet an der **Aufbauorganisation**, bietet es sich an, zwei Dimensionen zur Variation der Komplexität zu unterscheiden: Ihrer Gestalt wegen sollen diese Dimensionen mit vertikaler und horizontaler Komplexitätsvariation bezeichnet werden:

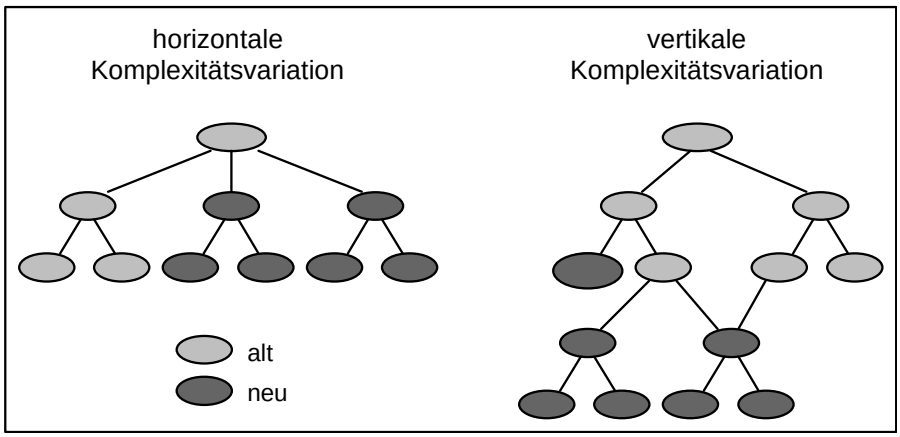



*Abbildung 9: Vertikale und horizontale Komplexitätsvariation*

Bei horizontaler Komplexitätsvariation wird die Zahl der Elemente und Verbindungen durch eine Verbreiterung oder Verschmälerung der Aufbauorganisation des Planspiels erreicht.

– Auf der Ebene bereits bestehender Systeme und Subsysteme werden parallel noch weitere Subsysteme gleicher oder ähnlicher Detaillierungstiefe (**Komplexität**) eingehängt oder entfernt. Als Beispiele könnte man sich das Hinzu-

fügen einer neuen Marktregion mit Marktteilnehmern und Produkten, eines weiteren Konkurrenzunternehmens oder eines neuen Produktes bei allen Unternehmen vorstellen. Diese Möglichkeiten sind mit Partialmodellen realisierbar.

Wesentlich weitergehende Möglichkeiten ergeben sich bei der Veränderung der Anzahl der Elemente und Verbindungen durch vertikale Komplexitätsvariation:

- Objekte können in ihrem Verhalten und ihren Vernetzungsmöglichkeiten verändert werden: Möglichkeiten hierfür sind beispielsweise die Veränderung des Protokolls durch Hinzufügen oder Entfernen von Eigenschaften und die Aktivierung oder Stillegung von Verhaltensfunktionen. Hinzu kommt die Möglichkeit, die Charakteristika der Verhaltensfunktionen durch Veränderung der Parameter, der stochastischen Komponente oder des Funktionstyps zu beeinflussen. Reichen die durch die Klasse vorgegebenen Variationsmöglichkeiten für das Objekt nicht aus, so kann - falls in der Objektbibliothek vorgesehen - auf ein Exemplar einer Subklasse zurückgegriffen werden, das erweiterte, aufwärtkompatible Möglichkeiten bietet.[202]

- Objekte, die bisher die feinste Modellierungsstufe darstellten, werden ersetzt durch Subsysteme mit noch detaillierterer Modellierung. Die Tiefe des Modells steigt. (Natürlich gilt dies auch für den entgegengesetzten Fall.)
  Dies ist nur möglich, wenn auch das Konzept der Ablaufmodellierung (vgl. 4.3.2) die Bildung von Hierarchien unterstützt. Ein Beispiel hierfür ist die Ersetzung der globalen Außendienstmodellierung in Form eines Wertes durch ein Submodell der Außendienstorganisation mit Filialen, Abteilungen und Mitarbeitern.

Mit jeder Variation der Komplexität der **Aufbauorganisation** geht eine Veränderung der qualitativen Dynamik - und damit der Ablauforganisation - einher: Neue Objekte werden in die Simulationsabläufe eingebunden, weggefallene Objekte entfernt.

Erstreckt sich die veränderte Dynamik auch auf die Komponente der Zeit - ändern sich also Auf- und Ablauforganisation dynamisch - so bringt dies eine weitere Dimension der Komplexitätsvariation ins Spiel, die die Entscheidungssituation der Teilnehmer beeinflußt.[203] Hierfür gelten die bereits bei der Diskussion über das Mikroverhaltensmodell angeführten Einschränkungen (vgl. 4.3.3).

---

202 Alle diese Variationen sind mit Object-VersPlan vom Anwender (Spielleiter) ohne Eingriffe in die Programmierung möglich. Beispiele hierfür finden sich im Abschnitt 5.3.1.

203 Zeitliche Veränderungen sind wesentlich schwieriger zu beherrschen als Veränderungen von Strukturen. *Dörner* führt dies auf die Probleme zurück, die Menschen bei der Erfassung der Dimension Zeit haben: Struktur und qualitative Zusammenhänge werden als Raumgestalten repräsentiert, in denen sich der Entscheider bewegt. Für die Erfassung der vierten Dimension - der Zeit - fehlt die Vorstellung von der Gestalt.
Vgl. Dörner, D. (Logik, 1993), S. 156 ff.

Doch auch die oben angesprochenen ‘einfacheren’ Möglichkeiten zur Komplexitätsvariationen fordern die objektorientierte Planspielentwicklung.

### Probleme

Eine Konsequenz aus der Komplexitätsvariation läßt sich mit dem ‘Problem der Aufwärtskompatibilität’ umschreiben.

Erste Voraussetzung für Aufwärtskompatibilität ist die Verfügbarkeit robuster Modell-Objekte in der Klassenhierarchie.[204] Die Forderung nach Robustheit läßt sich in zwei Kriterien konkretisieren:

– Die Protokolle der Klassen müssen dem Prinzip der Polymorphie folgen. Dann können Objekte leichter durch Objekte anderer Klassem ausgetauscht werden.

– Die Verhaltensmodellierung sollte so gewählt werden, daß Erweiterungen und Einschränkungen ohne weitergehende Eingriffe möglich sind.

Wenn detailliertere Submodelle einfachere Modellierungen ersetzen sollen, werden Mechanismen zur Aggregation benötigt. Sie sollen dafür sorgen, daß das neue Protokoll kompatibel mit dem der bisher letzten Ebene bleibt. Wenn dies der Fall ist, können die bisherigen Abläufe unverändert übernommen werden.[205]

Ein prinzipielles Problem mit detaillierteren Submodellen wurde bereits im vorangegangenen Abschnitt angesprochen: Die Kompatibilität von Mikro-und Makro-Modellierung. In der Praxis der Planspielkonstruktion könnte dies bedeuten, daß Submodelle, die dem Mikro-Ansatz folgen, lediglich mit unverändertem Protokoll als Ersatz für bestehende gröbere Modelle verwendet werden - und nicht mit zusätzlichen Verbindungen in die Dynamik eingreifen.

Ein weiteres Problem wurde ebenfalls schon mehrfach angesprochen: Die Sicherstellung der Wiederverwendung: Im Fall der Planspielentwicklung heißt dies, daß die Elemente der Objektbibliothek so beschaffen sein müssen, daß sie in verschiedenen Anwendungsfällen benutzt, durch die Kombination mit anderen Objekten zu wiederverwendbaren Komplexen (Submodelle, beispielsweise Unternehmen) angeordnet und gegebenenfalls durch Spezialisierungen erweitert werden können. An dieser Stelle soll nicht die schon ausführlich geführte Diskussion über die technischen Aspekte der Wiederverwendung wiederholt werden. Der Objektorientierte Ansatz hat hierfür seine Tauglichkeit bereits vielfach bewiesen.[206] Es soll vielmehr eine anwendungsnäheres Problem der Wiederverwendung angesprochen werden:

---

204 Ein Beispiel findet sich in der Model-Hierarchie von Buchalter, Auktionator und Versicherungs-Auktionator (vgl. 5.1.1 und 5.3.2 )

205 Objekt-VersPlan bietet für die Aufgabe der Konsolidierung die Klasse der Buchhalter an (vgl. 5.1.1).

206 Stellvertretend sei verwiesen auf: Broer, H. (Software-Montagetechnik, 1994), S. 68 ff.

– Wenn unterschiedliche Ersteller mit der Implementierung eines Problems beschäftigt sind, so werden schon allein durch deren unterschiedlichen Kontext und die unterschiedliche Wahrnehmung des abzubildenden Wirklichkeitsausschnittes verschiedenartige, inkompatible Lösungen entstehen. Deshalb werden für Planspiele aus unterschiedlichen Anwendungsbebieten weitgehend vollständige Klassenbibliotheken mit aufeinander abgestimmten Klassen benötigt - sogenannte Frameworks. Erst dann können die hier angesprochenen Möglichkeiten zur Komplexitätsvariation auch wirklich genutzt werden.

Moderne Lerntheorien, wie sie beispielsweise im Cognitive Apprenticeship-Ansatz (vgl. 3.1.1) umgesetzt wurden, fordern Möglichkeiten zur situationsspezifischen Komplexitätsanpassung - am besten, während das Planspiel gerade läuft. Deshalb ist die Verfügbarkeit der gerade beschriebenen Lösungsansätze eine wichtige Voraussetzung zum Erfolg der objektorientierten Planspielentwicklung - Object VersPlan, das im fünften Kapitel Gegenstand der Betrachtungen sein wird, bietet bereits funktionierende Lösungen für die meisten der hier beschriebenen Probleme an.

„If you don’t think your customer
is the most important part of your day -
think again“[207]

## 4.4. Ein qualitätsorientiertes Konzept für die objektorientierte Entwicklung von Planspielen

Die vorangegangenen Abschnitte des vierten Kapitels haben die objektorientierte Methodik und Technik als einen Ansatz zur Bewältigung der aus theoretischen Überlegungen und praktischen Erfahrungen definierten Anforderungen an die Planspielentwicklung diskutiert und den klassischen Techniken und Lösungsansätzen gegenübergestellt.

Jetzt soll der Anwender - der Kunde - im Vordergrund der Betrachtungen stehen: Wie können die mit Hilfe des Ansatzes der objektorientierten Planspielentwicklung erarbeiteten Konzepte und Lösungen in der Praxis eingesetzt werden? Im folgenden wird deshalb versucht, die theoretische Basis und die Probleme der Projektierung von Planspielen in einem qualitätsorientierten Entwicklungsmodell zusammenzuführen.

- Es ist notwendig, sich mit der Position der Qualitätsorientierung vertraut zu machen.
- Die Aufgaben der Planspielentwicklung können vor dem Paradigma der Qualitätsorientierung in ein Vorgehensmodell eingeordnet werden.
- Objektorientierung verändert die Organisationsformen der Planspielentwicklung.

### 4.4.1. Die Position der Qualitätsorientierung

#### Was ist Qualität?

Gibt es eine eindeutige Definition des Qualitätsbegriffs? Den Überlegungen von *Helten* / *Schmidt* / *Schneider* zur Qualität folgend, gibt es keinen streng normativen Ansatz bei der Qualitätsdefinition:[208] Außer erkenntnistheoretischen Überlegungen - wie sie im

207 Warth, W. P. (Think, 1992), S.3

208 Helten, E.,Schmidt, H., Schneider, W. (Qualitätszirkel, 1992), S. 999

Abschnitt 3 für Planspielentwicklung angestellt wurden (Anm. d. Verf.) - kann im Sinne des Individualbedürfnis-Postulats und dem zunehmenden Wertepluralismus[209] kein allgemeingültiger Katalog von Qualitätskriterien aufgestellt werden: Jeder Kunde definiert seine Ansprüche an die Leistung für sich individuell.

Es muß vielmehr Sorge dafür getragen werden, daß ein Ansatz zur Verfügung steht, der ausreichende Flexibilität und Mächtigkeit besitzt, um den unterschiedlichen Kundenbedürfnissen Rechnung tragen zu können.

Ein Planspielentwicklungsprogramm für Planspiele aller Art kann deshalb nicht angestrebt werden: Durch die Konzentration beispielsweise auf die Erstellung einer Klassenbibliothek für Unternehmensplanspiele aus dem Bereich Versicherungswirtschaft kann viel eher ein hoher Qualitätsstandard aus Sicht des Kunden gewährleistet werden.

### Methodische Grundlage: Kunden/Lieferanten-Beziehungen

Grundlegende Idee für den im folgenden vorgestellten Ansatz der Qualitätsorientierung in der Planspielentwicklung ist, die Beziehungen von Kunden und Lieferanten in den Mittelpunkt der Betrachtung zu rücken. Von Kunden und Lieferanten wird immer dann gesprochen, wenn jemand eine Leistung für einen anderen erbringt[210]; gleichgültig, ob es sich dabei um einen Partner innerhalb oder außerhalb der Organisation (des Unternehmens oder der Bildungseinrichtung ...) handelt. Oberstes Maß für die Leistungserfüllung ist die Zufriedenheit des Kunden mit der erbrachten Leistung. [211]

### Kunden und Lieferanten bei der Planspielentwicklung

Kunden/Lieferanten-Beziehungen nehmen für die Qualitäsorientierung eine Schlüsselposition ein - wer sind also die Kunden und Lieferanten bei der Planspielentwicklung?

209 Vgl. Lehmann, A. (Dienstleistungsmanagement, 1993), S. 15 f.

210 Mit Kunde ist derjenige gemeint, bei dem ein Problem liegt, nicht derjenige, der den Auftrag erteilt. Vgl. Wolff, F. (Total Quality, 1994), S. 41

211 Vgl. Lehmann, A. (Führungsdimension, 1989), S. 664;
vgl. a. Schmidt, H., Hardt, M. (Strukturwandel, 1994), S. 1433 ff.

Die folgende Abbildung erarbeitet die Beziehungen:

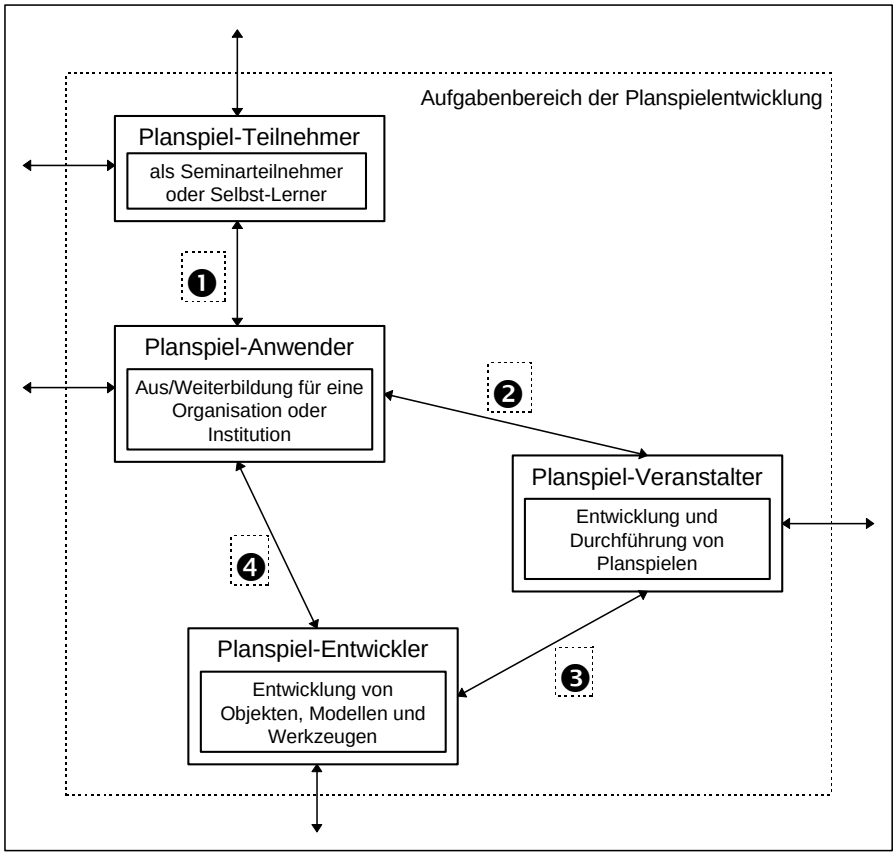



*Abbildung 10: Kunden und Lieferanten der Planspielentwicklung*

Die Teilnehmer dieses Netzwerks zur Planspielentwicklung sind in mehrfacher Hinsicht Kunde und Lieferant gleichzeitig:

❶ Planspiel-Anwender und Planspiel-Teilnehmer sind füreinander Kunde und Lieferant. Die Teilnehmer werden als Kunde vom Planspiel-Anwender mit Aus- und Weiterbildung versorgt - im Gegenzug liefern sie ihrer Bildungsorganisation (hier: Planspiel-Anwender) die notwendigen Informationen für die Ausarbeitung eines passenden Bildungskonzepts.

❷ Anwender beziehen ein Planspiel-Produktbündel vom Planspielveranstalter. Andererseits liefern sie aber den Veranstaltern die notwendigen Informationen und Einblicke in die Bedürfnisse und Erfordernisse von Bildungsorganisationen.

❸ Ein Planspiel-Veranstalter ist in der Regel kein Spezialist für die Umsetzung von **Simulationsmodellen** per Computer. Er bezieht deshalb die für ein Planspiel

notwendige Software von einem Softwareentwickler. Ohne intensive Zusammenarbeit und Erfahrungsaustausch mit dem Planspielveranstalter kann ein Entwickler keine anforderungsgerechten Softwarebausteine und Werkzeuge zur Verfügung stellen.

❹ Damit die Objekte für spezialisierte Anwendungsfälle den Bedingungen und Anforderungen der Anwender entsprechen, sind Planspiel-Entwickler und Anwender in engem Kontakt.

Es liegt auf der Hand, daß nicht in allen Fällen der Planspielentwicklung alle Partner in der hier vorgestellten Form zusammenwirken: Es ist durchaus denkbar, daß beispielsweise die Funktionen des Entwicklers und des Veranstalters zusammenfallen. Die Diskussion der organisatorischen Aspekte wird im Abschnitt 4.4.3 vertiefend aufgegriffen.

### Das Planspiel als Dienstleistung?

Aus den bisherigen Ausführungen sollte klar geworden sein: Ein Planspiel ist mehr als nur ein Simulationsprogramm auf dem Computer. Es ist vielmehr ein Bündel, das für die Bedürfnisse der verschiedenen Kunden individuell geschnürt wird - ein Bündel von aufeinander abgestimmten Gütern und Dienstleistungen[212]:

- Das pädagogische Lehr/Lernkonzept, das sich an den Bedürfnissen der Planspiel-Anwender orientiert.

- Das Simulationsmodell, das die Basis für die Computerunterstützung zur Durchführung des Planspiels liefert. Kontextbezogene Objektbibliotheken ermöglichen die Anpassung an die spezifischen Erfordernisse der Einsatzsituation.

- Die Werkzeuge, die den Anwender bei Erstellung und Durchführung von Planspielsimulationen unterstützten.

- Die Durchführung des Planspiels durch Trainer, die mit den pädagogischen Konzepten und dem fachlichen Kontext gleichermaßen vertraut sind.

Vergleicht man diese - zugegebenermaßen zukunftsorientierte - Auffassung von Planspielen mit den heute am Markt verfügbaren Angeboten, so ist deutlich die fehlende Möglichkeit zum ‘unbundling’ von Leistungen festzustellen:

212 An dieser Stelle sei auf die überaus kontrovers geführte Diskussion über den Charakter von Dienstleistungen verwiesen. So führen *Stauss* / *Hentschel* eine Vielzahl von Definitionsversuchen und Abgrenzungen zum Thema Dienstleistungen auf.
Vgl. Stauss, B., Hentschel, B. (Dienstleistungsqualität, 1991), S. 238 f.

– Buchen von Seminarplätzen: Wenn bisher ein Unternehmen (oben mit ‘Planspiel-Anwender’ bezeichnet) Mitarbeiter zu einem Planspielseminar schickt, so nimmt es das standardisierte Bündel der gesamten planspielrelevanten Dienstleistungen und Produkte in Anspruch - eine individuelle Anpassung an die Wünsche eines Kunden kann nicht stattfinden. Solche ‘offenen Planspiele’ sind nur dann geeignet, wenn es um die Vermittlung allgemeiner **Lernziele** wie **soziale Kompetenz**, **Systemkompetenz** oder allgemeinen Fachinhalten geht. Doch bereits dieser recht unspezifische Mix angebotener Lernziele macht ein solches Schulungsangebot nur für einen relativ kleinen Ausschnitt von Teilnehmern wirklich interessant.

– Kauf von Planspiel-Software: Am Markt ist eine Reihe vorgefertigter ‘Planspiele’ als Standardsoftware verfügbar. Die Anpassungmöglichkeiten an spezifische Lernumgebungen und -ziele sind in der Regel sehr beschränkt - mehr als ein Drehen an Parametern und Startdaten ist meist nicht möglich. Auch adressiert jedes dieser Planspiele eine spezielle Zielgruppe: Kontext und Komplexität sind fest vorgegeben und nicht beeinflußbar.

  Kaufplanspiele eignen sich am ehesten für Selbstlernumgebungen zur Vermittlung standardisierter Themen - das Angebot ist in diesem Segment vor allem aus der Unterhaltungsbranche gewachsen. Einer guten grafischen Aufmachung stehen oft didaktische Schwächen und starre Lerninhalte gegenüber. Die Kosten sind wegen der hohen Multiplikatorwirkung in der Regel sehr gering.

  Standard-Planspiele[213], die sich auf die Unterstützung der klassischen Planspielsituation konzentrieren, werden oft als Basis für die Weiterentwicklung zu einem ‘individuellen’, organisationsintern veranstalteten Planspiel verwendet: Sie leiden jedoch auch unter den bereits mehrfach angeführten Beschränkungen konventioneller (=nicht objektorientierter) Planspielentwicklungen. Werden unter hohen Aufwendungen Anpassungen gemacht, die den Einsatz innerhalb der Organisation an den dort benötigten Lernkontext verbessern, so sind sie bestenfalls so ‘gut’, wie ein freies Planspiel, dessen Zielgruppe exakt besetzt wurde.[214]

– Entwicklung eines individuellen Planspiels: Wenn kein passendes Standard-Planspiel am Markt verfügbar ist - und das ist bei gesteigerten Qualitätsansprüchen wohl zu erwarten - muß auf Individualentwicklung zurückgegriffen werden: Hier stellt sich für den Anwender die Entscheidung über Eigenfertigung oder Fremdbezug. Da die meisten Anwender nicht über Know-How zur Planspielkonstruktion verfügen und dies auch nicht selbst vorhalten oder aufbauen möchten, greifen Sie auf die Angebote von

213 Eine Übersicht europäischer Planspiele findet sich in: Rohn, W. (Planspiel-Übersicht, 1992)

214 So hat beispielsweise die Allianz AG eine Version des in dieser Arbeit bereits mehrfach zitierten Versicherungsplanspiels von *Hansen* eingekauft und mit hohem Aufwand um einige wenige Merkmale erweitert, die es den Spezifika dieses Unternehmens annähern. Damit ist jedoch das Entwicklungspotential eines derartigen Planspiels weitgehend ausgereizt.

Planspielkonstrukteuren zurück. Unternehmen wie UNICON, die sich auf Planspiele konzentriert haben, treten als Entwickler und als Veranstalter von Planspielen auf.[215] Die Kosten für eine solche Lösung werden naturgemäß höher ausfallen als für die Verwendung eines bestehenden Planspiels. Bedingt durch das konventionelle Design des Simulationsmodells erkauft man sich aber trotzdem - analog zum zuvor beschriebenen Fall - nur maximal eine Lösung für einen spezifischen Anwendungsfall.

Und was haben diese bisher am Markt verfügbaren Lösungen mit der heute geforderten Ausbildungsqualität zu tun? Man möchte provokanterweise sagen: Nicht viel. Man kann wohl davon ausgehen, daß die Kunden sich von den verfügbaren Leistungen, die für Ihre Bedürfnisse am ehesten geeigneten aussuchen werden - doch sollen bei dem für die Zukunft so wichtigen Thema der Erhaltung und Förderung des Potentials der Mitarbeiter die Weiterbildungsbedürfnisse soweit reduziert werden, bis sie optimal auf das Angebot passen? Die optimale Vorbereitung der Lernenden auf die Herausforderungen zukünftiger Problemstellungen (vgl. 3.1.1) können solche Lösungen nicht leisten. Objektorientierte Planspielentwicklung tritt mit dem Anspruch an, Qualität im Sinne der „Erfüllung der Bedürfnisse des Kunden“[216] zu begreifen.

Den Weg dorthin weist das objektorientierte Vorgehensmodell, wie es im folgenden Abschnitt vorgestellt wird.

215 Vgl. Högsdal, B. (Planspiele, 1992), S. 83 ff.

216 Helten, E.,Schmidt, H., Schneider, W. (Qualitätszirkel, 1992), S. 998 f.

### 4.4.2. Ein objektorientiertes Vorgehensmodell der Planspielentwicklung

Basierend auf den eingangs beschriebenen Kunden/Lieferanten-Beziehungen läßt sich ein Vorgehensmodell für die Planspielentwicklung ableiten:

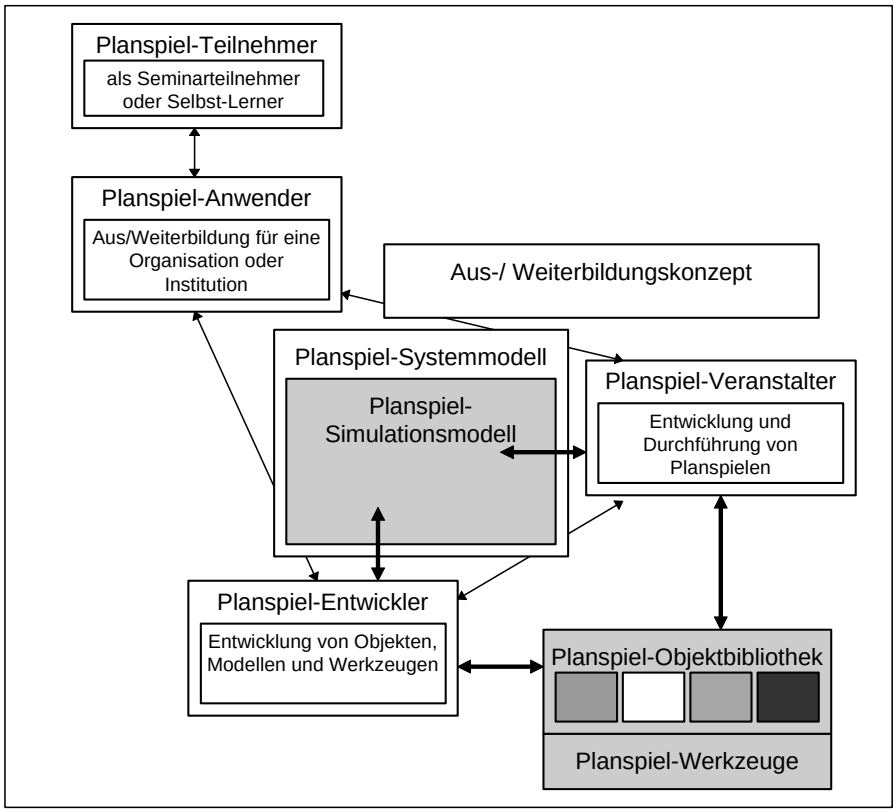



*Abbildung 11: Vorgehensmodell der Planspielentwicklung*

Die Abbildung zeigt, wie das objektorientiert formulierte Planspiel-Systemmodell die gemeinsame Basis und Dokumentation für die Auseinandersetzung mit der Aufgabe der Planspielentwicklung wird. In ihm werden alle relevanten Elemente eines Planspiels beschrieben - gleichgültig, ob sie später durch Software implementiert werden sollen oder nicht. Diese Implementierung findet sich im Simulationsmodell, daß vom Veranstalter und/oder vom Entwickler nach den Anforderungen des Anwenders konstruiert wird. Basis der Planspielkonstruktion ist eine Objektbibliothek (vgl. 5.1) und ein Satz von Werkzeugen (vgl. 5.2). Man erkennt, daß objektorientierte Planspielentwicklung kein linearer Ablauf, sondern vielmehr

ein iterativer Prozeß mit Rückkoppelungen innerhalb der Kunden-Lieferanten-Beziehungen ist.[217]

Damit die Vorgehensweise vom Problem zur Implementierung deutlich wird, soll der Entwicklungprozeß in drei Aufgabenbereiche zerlegt werden: Analyse, Design und Implementierung.[218] Klassische objektorientierte Vorgangsmodelle wie OMT legen den Schwerpunkt auf die einheitliche Semantik und die Dokumentation des Entwicklungsprozesses - eine Aufgabe, die mit Hilfe der schon mehrfach angesprochenen Werkzeuge und der Klassenbibliothek bereits erledigt werden kann. Ein nach Meinung des Autors viel wichtigerer Punkt wird von diesen Verfahren jedoch vernachlässigt: Die Kommunikation zwischen Kunden und Lieferanten.[219]

### Mittel zur Kommunikation zwischen Kunde und Lieferant

Der objektorientierte Ansatz stellt von sich aus Konzepte zur Verfügung, die die Kommunikation zwischen Kunden und Lieferanten unterstützen: Durch die Verringerung der semantischen Lücke kann mit allen Beteiligten auf einem problemorientierteren Sprachniveau kommuniziert werden. Eine Qualitätssicherung ist durch das ständige Wechselspiel zwischen Kunde und Lieferant fest im System eingebaut: Entspricht etwas nicht den Erwartungen, kann der Kunde dem Lieferanten dies noch während der Entwicklungsarbeit mitteilen. So berichtet beispielsweise *Thelen* von Teilnehmern einer Konferenz über objektorierte Softwareentwicklung, die die Forderung nach einem Lösungskonzept in den Mittelpunkt der Diskussion gestellt haben, bei dem Auftraggeber und Auftragnehmer die selbe Sprache sprechen.[220]

Mächtige, interaktive Entwicklungswerkzeuge, orientiert an den Bedürfnissen der Benutzer, in Verbindung mit wiederverwendbaren, anwendungsspezifischen Komponenten können - entsprechend angepaßt - im Abstimmungsprozess über die benötigten Leistungen von beiden Seiten verwendet werden. Ein erster Lösungsansatz kann bereits in gemeinsamen Gesprächen erarbeitet und diskutiert werden: Die wesentlichen Ideen dieser Vorgehensweise lassen sich unter Rapid Prototyping und Wiederverwendung zusammenfassen.

Objektorientierung unterstützt ein ganzheitliches Verständnis von den Problemen: Nicht alle Elemente der Aufgabe müssen später mit Hilfe der (objektorientierten) Software-

---

217 Das Aus- und Weiterbildungskonzept spielt eine entscheidene Rolle bei der Formulierung der Lernziele und damit für die Formulierung der Anforderungen an das Planspielkonzept. Es soll hier keine Diskussion über die Problematik der Überführung von Zielen der Aus- und Weiterbildung geführt. Hierzu sei beispielsweise verwisen auf: Schmidt, H. (integratives Konzept, 1993), S. 8 ff.

218 Diese Dreiteilung findet sich in beinahe allen Veröffentlichungen zur objektorientierten Softwareentwicklung.
Einen Überblick über die Methoden bietet beispielsweise: Stein, W. (Vergleich, 1993), S. 317ff

219 Vgl. Wolff, F. (Total Quality, 1994), S. 40 ff.

220 Vgl. Thelen, B. (Ordnung, 1994), S. 81

technik implementiert werden, wenn besser eine andere Lösung angezeigt ist. Mit den modernen (im Gegensatz zu den oben angesprochenen klassischen) Methoden CRC (Class, Responsibilities, Collaboration von *Beck* / *Cunningham*), RDD (Responsibility Driven Design, *Wirfs-Brook* / *Wilkerson*) und OBA (Object Behaviour Analysis, *Rubin* / *Goldberg*) stehen Konzepte zur Verfügung, die sich zuerst an der Aufgabe orientieren.[221]

- In der Analysephase entsteht ein Systemmodell mit Objekten: Statt jedoch bereits jetzt Objektattribute (Eigenschaften) zu beschreiben, werden zuerst die Verantwortlichkeiten (engl. responsibilities) aufgeführt: Ob und wie sie durch Softwaretechnik umgesetzt und implementiert werden, wird erst in einer späteren Phase entschieden.

- Verantwortlichkeiten können mit Hilfe von Szenarien aufgespürt werden: Anhand typischer Abläufe werden die Objekte und ihre Rolle identifiziert und ihr notwendiges Verhalten beschrieben.

Ideen dieser Methoden könnten auch in Planspiel-Entwicklungsprojekten eingesetzt werden, die so spezifisch sind, daß von den Objektbibliotheken lediglich die allgemeinen Simulationsobjekte verwendet werden können und spezialisierte Objekte in größerem Umfang neu entwickelt werden müssen.[222]

### 4.4.3. Neue Möglichkeiten der Organisation der Planspielentwicklung

Wie bereits im vorangegangenen Abschnitt angeklungen ist, liegt eines der Probleme der Planspielentwicklung in der Tatsache begründet, daß Organisationen, die Planspiele nutzen wollen, einen eingeschränkten Gestaltungsspielraum bei der Organisation des Entwicklungsprozesses haben: Entweder sie beziehen standardisierte Leistungen vom Markt, oder sie erstellen selbst ein Planspiel, daß sich näher an ihren Bedürfnissen und denen ihrer Kunden orientiert. Doch, „Wer im zukünftigen Wettbewerb glaubt, alles selbst machen zu müssen, der verzichtet auf die Vorteile der arbeitsteiligen Industrie- und Dienstleistungsgesellschaft und verschwendet knappe Ressourcen.“[223]
Diese Mahnung trifft sicherlich auch für die Kunden von Unternehmensplanspielen zu.

---

221 Vgl. hierzu und zum folgenden: Thelen, B. (Ordnung, 1994), S. 81

222 Für eine weiterführende Diskussion fehlt hier der Raum. Gerade an diesem Punkt ist auch der Nachholbedarf von Seiten des Softwareengineerings noch enorm. Man sieht dies an der kontrovers und hartnäckig geführten Diskussion, welches wohl die objektorientierte Programmiersprache und die Entwicklungsmethode der Zukunft sein werden. In Sinne der Qualitätsorientierung ist diese Diskussion fruchtlos: Es wird sich die Methode durchsetzen, die den Kunden die besten Ergebnisse bei der Bewältigung ihrer Aufgaben bringt. In Zeiten schnelleren Wettbewerbs und immer komplexer werdender Strukturen wird sich letztendlich derjenige durchsetzen, der die Kommunikationsprobleme - eben die Kunden/-Lieferantenbeziehungen - besser und schneller löst.

223 Picot, A., Gerhardt, T., Nippa, M. (Leistungstiefenentscheidungen, 1992), S. 1

Die folgende These soll einen Ausweg aus der Dichotomie von Eigenfertigung oder Fremdbezug andeuten:

‘Objektorientierte Planspielentwicklung erhöht die Flexibilität bei der Gestaltung der Leistungstiefe’

Um sich dieser Ausage zu nähern, lohnt ein Blick auf die Entscheidung über die Gestaltung der optimalen Leistungstiefe - die Frage nach der „Verteilung von Teilleistungen zwischen Unternehmen und Markt“[224]. Diese Frage stellt sich prinzipiell für alle drei angeführten Beteiligten an der Planspielentwicklung: Den Anwender, den Veranstalter und den Entwickler.

Um die optimale Leistungstiefe für diese unterschiedlichen Kunden bestimmen zu können, bedarf es einer tragfähigen Methode. Grundsätzlich kann festgehalten werden, daß man beim Einsatz marktnäherer Organisationsformen der Koordinationsproblematik verstärkte Aufmerksamkeit schenken muß. Deshalb schlagen *Picot* / *Gerhard* / *Nippa* eine an den Transaktionskosten orientierte Entscheidung über die Leistungstiefe vor.[225] Als Substitut für die in der Praxis nicht meßbaren Transaktionskosten werden dabei die Charakteristika der Leistung herangezogen und operationalisiert: Spezifität, strategische Relevanz und Unsicherheit. Jedoch können Barrieren die optimale Gestaltung behindern oder verzögern; fehlendes Know-How oder mangelnde persönliche Flexibilität sind Beispiele hierfür.

Was spricht eher für die Erhöhung des Anteils der Eigenfertigung bei der Entwicklung von Planspielen? *Picot* / *Reichwald* sehen eine Leistung (Aufgabe) dann als potentiellen Kandidaten für Eigenerstellung, wenn jene spezifisch und gleichzeitig strategisch bedeutsam ist. Sie führen als ein Beispiel hierfür explizit die Beschäftigung mit dem Humankapital einer Organisation an - also die Aus- und Weiterbildung der Mitarbeiter zum Ausbau und der Festigung eines Wettbewerbsvorteils. Zur Vermeidung des damit einhergehenden innerbetrieblichen Abhängigkeitsverhältnisses empfehlen sie integrative Koordinationsformen[226]. Die in den vorangegangenen Absätzen beschriebene Betrachtung von Kunden/Lieferanten-Beziehungen vor dem Hintergrund von Qualitätsaspekten könnte ein Ansatz hierfür sein.

---

224 Picot, A., Gerhardt, T., Nippa, M. (Leistungstiefenentscheidungen, 1992), S. 2

225 Vgl. hierzu und zum folgenden: Picot, A., Gerhardt, T., Nippa, M. (Leistungstiefenentscheidungen, 1992), S. 4 ff.
Die Autoren entwerfen Normstrategien für die Leistungstiefengestaltung, die in den folgenden Beispielen zur Planspielentwicklung angewendet werden.
Neben dem Transaktionskostenansatz als Entscheidungskalkül über die Leistungstiefe führen *Picot* / *Franck* zur Frage der vertikalen Integration Ansätze der Industrial Organisation -Theorie und Produktionskostenüberlegungen an.
Vgl. Picot, A., Franck, E. (Vertikale Integration, 1993), S. 183 ff.

226 Vgl. Picot, A., Reichwald, R. (Kooperationsformen, 1994), S. 5

Diese Situation beschreibt diejenige eines Unternehmens, das die Aus- und Weiterbildung seiner Mitarbeiter als spezifischen Wettbewerbsvorteil betrachtet und deshalb als Planspiel-Anwender zur Eigenerstellung tendiert. Unterstützt wird diese Haltung, falls das Unternehmen häufig individuelle Planspiel-Varianten benötigt. Für Anwender, deren Kerngeschäft in der Regel nichts mit Planspielen und Planspielentwicklung zu tun hat, bauen sich aber Barrieren in Form von Know-How-Defiziten auf. Objektorientierte Planspielentwicklung kann in diesem Fall zu einer Verschiebung der Gewichte durch Reduzierung der Barrieren führen: Die allgemein verwendbaren Objekte und Werkzeuge können - so am Markt angeboten - fremdbezogen werden, Objekte hoher Spezifität werden selbst erstellt oder als Individualsoftware in Auftragsarbeit konstruiert. Doch es bleibt noch die hohe strategische Bedeutung: Nur, wenn der Kauf der Teile und Werkzeuge auch das Recht zur uneingeschränkten, exklusiven Verwendung (und nicht nur zur Benutzung) und eine entsprechende Dokumentation beinhalten, wird sich ein Unternehmen auf Fremdbezug einlassen. Objektorientiertes Design und entsprechende Werkzeuge erleichtern das Nachvollziehen der Entwurfsprinzipien zugekaufter Elemente erheblich. In Zeiten großer Unsicherheit, wie sie beispielsweise momentan in dem im Deregulierungsprozeß befindlichen Versicherungsmarkt beobachtet werden kann, neigen Planspiel-Anwender eher zur Eigenfertigung.

Für Planspiel-Veranstalter ist die Verfügbarkeit eines überlegenen Planspiels - bestehend aus Konzept, Durchführung und Computerunterstützung - das entscheidende Argument im Wettbewerb. Er wird danach trachten, einen vorhandenen Vorsprung als Barriere gegen Wettbewerber aufrecht zu erhalten. Strategische Partnerschaften und Kooperationen könnten Koordinationsformen sein, die einerseits den strategischen Bedenken und den Problemen der Unsicherheit der Leistung begegnen können und andererseits die gewünschten Barrieren zumindest mittelfristig sichern. Zudem ist dies eine Lösung, die auch kleineren Planspiel-Anwendern als Kunden offensteht. Doch der Veranstalter steht selbst vor ähnlichen Problemen wie der Anwender: Er ist auf die für bestimmte Anwendungskontexte spezifische Leistung des Entwicklers angewiesen: Daß die im Abschnitt 2.3 diskutierten Probleme der Standardisierung von Planspielobjekten überwunden werden können, ist nicht zu erwarten. Die strategische Zielrichtung für den Veranstalter geht deshalb ebenfalls ganz klar in Richtung Eigenfertigung. Er wird anstreben, die vom Entwickler aufgebauten Barrieren abzubauen. Zwischenformen der Organisation sind wegen der verbesserten Kommunikationsmöglichkeiten objektorienierter Lösungen (vgl. 4.4.2) sehr viel eher denkbar, als es früher mit konventionelle Technik und Methodik der Fall war.

Bleibt noch der Planspiel-Entwickler: Aus seiner Sicht sieht die Betrachtung ein wenig anders aus. Für ihn, der von der Anwendung am ‘weitesten’ entfernt ist, ist die strategische Bedeutung und Spezifität der Leistung nicht so hoch wie für die anderen beiden Beteiligten. Er wird versuchen, seinen Know-How-Vorsprung zu sichern und auszubauen, und, falls dieser langfristig nicht zu halten ist, möglichst teuer zu verkaufen. Planspiel-Entwickler haben zwar während der Entwicklung die Gelegenheit, anwendungsspezifisches Know-how anzusammeln - es fehlt ihnen jedoch in aller Regel die

praktische Erfahrung und der theoretische Hintergrund, um eine realistische Chance für die Steigerung der Leistungstiefe zum Planspiel-Veranstalter zu haben.

Dynamische Märkte fordern nach flexiblen Organisationsformen. Die Rollen von Anbietern und Nachfragern wechseln ständig. Der objektorientierte Ansatz kann dazu beitragen, die notwendige Kommunikation im Fall der Planspielentwicklung in Gang zu setzen.

Das folgende fünfte Kapitel verläßt nun die theoretische Ebene und versucht, zu zeigen, wie die bisher erarbeiteten Konzepte und Ideen in der Praxis umgesetzt werden können.

"It's a holy grail.
There is no panacea." [227]

# 5. Objektorientierte Entwicklung von Unternehmensplanspielen mit Object-VersPlan

Ein wesentliches Ziel der vorliegenden Arbeit ist es, die Anwendbarkeit der zuvor erarbeiteten und theoretisch fundierten Konzepte nachzuweisen. Deshalb ist mit Object-VersPlan eine objektorientierte Planspielentwicklungsumgebung entstanden, die konsequent dem objektorientierten Paradigma folgend implementiert wurde.

Herzstück von Object-VersPlan ist die Klassenbibliothek, die die allgemeinen, wiederverwendbaren Beschreibungen (=Klassen) für die Bestandteile objektorientierter Unternehmensplanspiele im Bereich Versicherungswirtschaft enthält. Sie wird im ersten Abschnitt vorgestellt. Dabei wirken neben Erkenntnissen aus den Disziplinen der angewandten Informatik und der Systemforschung auch die Fachbereiche Versicherungswirtschaft, Organisationslehre und Organisationspsychologie zur Repräsentation des domänenspezifischen (=für das Anwendungsgebiet spezifischen) Wissens zusammen.

Der zweite Abschnitt des fünften Kapitels widmet sich den Werkzeugen, die dem Anwender die Erstellung, das Testen und die Durchführung von Planspiel-**Simulationsmodelle**n auch ohne Programmierkenntnis ermöglichen.

Im dritten Abschnitt wird anhand eines Beispiels die Anwendung der Möglichkeiten von Object-VersPlan vorgestellt.

227 Ausspruch von *Stroustrup* auf einer Konferenz für objektorientierte Softwareentwicklung. Er meint damit die Vorgehensweise bei objektorientiert angelegten Projekten: Identifizierung und Klassifizierung der Objekte für die Lösung eines Anwendungsproblems kann nicht mit Patentrezepten bewältigt werden.
Zitiert in: Booch, G. (OOD, 1991), S. 132

"An omnipresent problem in science is to
construct meaningful classifications
of observed objects or situations.
Such qualifications facitate human comprehension
of the observations and the subsequent development
of scientific theory."[228]

## 5.1. Eine Objektbibliothek für ein Versicherungsplanspiel

Die folgenden Abschnitte geben einen Einblick in den in Object-VersPlan vorgeschlagenen Ansatz zur Lösung der von *Michalski*, *Stepp* und *Stroustrup* angesprochenen Probleme mit der Abbildung eines relevanten Wirklichkeitsausschnittes. Sie sind das Ergebnis der im vorangegangenen vierten Kapitel erarbeiteten Methoden und Vorgehensweisen für objektorientierte Planspielentwicklung; an gegebener Stelle wird auf die zugrundeliegenden Konzepte verwiesen.

1. Im ersten Abschnitt wird die Abbildung der allgemeinen Beschreibungen der Systemelemente in einer Klassenhierarchie beschrieben, mit der die Mikrowelt für ein Versicherungsplanspiel modelliert werden kann.

2. Der zweite Abschnitt erklärt das Simulationskonzept.

3. Der darauf folgende Abschnitt stellt das **Metamodell** vor, das die Beziehungen der Systemelemente abbildet.

### 5.1.1. Klassenhierarchie

Nachdem im vierten Kapitel Lösungen zur Bewältigung der Problemdimensionen der Planspielentwicklung erarbeitet wurden, soll nun gezeigt werden, wie eine konkrete Lösung aussehen könnte: Für Object-VersPlan ist ein Prototyp einer wiederverwendbaren Klassenbibliothek entstanden, die Anworten auf die Anforderungen anwendungsbestimmter Planspielentwicklung liefert.

---

228 Michalski, R., Stepp, R. (Observation, 1983), S. 332

### Vom Problem zur Klassenhierarchie

Am Ende der Modellbildungsprozesse objektorientierter Analyse und Design steht die allgemeine Beschreibung der am Problem beteiligten Systemelemente (vgl. 4.1.1).

Es entsteht eine Klassenhierarchie mit allgemeinen und spezialisierten Klassen. Im vorliegenden Fall wurde diese Hierarchie mit Smalltalk/V als wiederverwendbare Klassenbibliothek implementiert und kann damit für viele Anwendungsfälle von Versicherungsplanspielen benutzt werden.

Die Klassenhierarchie fügt sich in ein bestehendes Basissystem einer objektorientierten Entwicklungsumgebung ein und macht von den Möglichkeiten bestehender Klassen Gebrauch.[229]

### Model-View-Controller

Die Verfolgung des M-V-C-Ansatzes (vgl. 4.3.3) trägt wesentlich dazu bei, die Vorteile der Objektorientierung zu nutzen.

Deshalb wird als Gliederungskriterium die Unterscheidung in Modell-Klassen, View-Klassen und Controller-Klassen gewählt.

229 Im Falle von Object-VersPlan basiert die Klassenbibliothek auf den Basisklassen des Images von Digitalks' Smalltalk/V und den Erweiterungen Window-Builder Pro und Subpanes von ObjectShare und MathPack sowie BusinesGraph von G-Soft.

### Modell-Klassen

Die Modell-Klassen bilden die logische Ebene des Systemmodells.

*Tabelle 19: Klassenhierarchie der Modell-Klassen*

| Klassenbezeichnung |
|---|
| Netzwerkmodell |
| Teilehierarchie |
| Ablaufmodell |
| Bericht |
| Spiel |
| Netzwerkobjekt |
| Verbindung |
| aktive Verbindung |
| Simulationsobjekt |
| Buchhalter |
| Auktionator |
| Auktionator für Versicherungsprodukte |
| Unternehmen |
| Versicherungsunternehmen |
| Produkt |
| Risiken |
| Mitarbeiter |
| Aufgabe |
| Bestandsbearbeitung |
| Antragsbearbeitung |
| Schadenbearbeitung |
| Benutzer |
| Spieler |
| Spielleiter |
| Planspiel-Wurzel |

Die Subklassen von Simulationsobjekten bilden das Herzstück der Modell-Klassenhierarchie. Hier finden sich all die Klassen für Objekte, die Elemente eines Systemmodells für ein Unternehmensplanspiel sein können.

Allgemeine Simulationsobjekte sind nicht auf einen Anwendungskontext festgelegt. Sie eignen sich prinzipiell für alle Arten von Unternehmensplanspielen.

– Die Klasse Simulationsobjekt ist die allgemeine Beschreibung eines Systemelements. Simulationsobjekte können innerhalb der Umgebung von Object-VersPlan an Simulationen teilnehmen (Details hierzu finden sich im Abschnitt 5.1.2.) und haben darüberhinaus keine spezifischen Eigenschaften und Fähigkeiten. Alle weiteren Subklassen sind Spezialisierungen der Klasse Simulationsobjekt.

  Exemplare der Klasse Simulationsobjekte und ihrer Subklassen können als Teilehierarchie angeordnet werden:

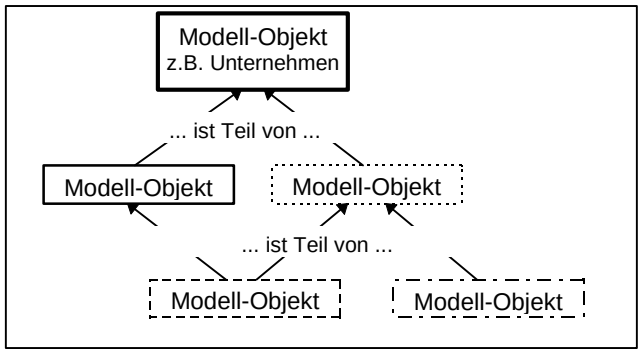



*Abbildung 12: Teilehierarchie*

  Durch eine Verbindung vom Typ ...ist Teil von... zwischen zwei Objekten entsteht eine Hierarchie in Form eines gerichteten Graphen: Objekt X ist Teil von Objekt Y. Es ist zulässig, daß ein Objekt Teilebeziehungen zu mehreren Objekten unterhält (Objekt X ist Teil von Objekt Y und Objekt X ist Teil von Objekt Z). Selbstverständlich kann ein Objekt beliebig viele Teile haben.

– Die Klasse Buchhalter implementiert allgemeine Fähigkeiten und Eigenschaften zur Durchführung einfacher Berechnungen, wie sie beispielsweise zur Konsolidierung von Werten benötigt werden. Von dieser Klasse können Spezialisierungen wie Bilanzbuchhalter (zur Führung von Konten bis hin zu Bilanz sowie Gewinn- und Verlustrechnung), Controller (zur Errechnung von Kennzahlen und dem Erstellen von Auswertungen[230]) oder auch Auktionatoren abgeleitet werden.

– Auktionatoren sind spezielle Buchhalter. Sie dienen der Koordination der Marktkräfte: Man kann sich Auktionatoren als eine Art unsichtbare Hand (engl. invisible hand) vorstellen, die die Pläne der Marktteilnehmer - Anbieter und Nachfrager - mit-

230 Diejenigen unter den Lesern, deren Beruf Controller ist, mögen dem Autor bitte diese Einordnung verzeihen.

einander abstimmt und letztendlich die Nachfragemengen und die Preise festlegt. Diese Marktauffassung steht dem Auktionatormodell von WALRAS nahe.[231]

Im Abschnitt 5.3.2 wird die Anwendung des Auktionators vorgestellt.

- Unternehmen bildet die abstrakte Klasse für Spiel-Unternehmen. Ohne Unternehmen ist selbstverständlich ein Unternehmensplanspiel nicht denkbar. Von ihr werden je nach Anwendungskontext die spezialisierten Unternehmen abgeleitet.

- Produkt ist ebenfalls eine abstakte Klasse. Sie bildet eine Menge von Produkten ab, für die beispielsweise die Eigenschaft des Durchschnittspreises modelliert ist.

- Eine wichtige Klasse für ein Unternehmensplanspiel ist die der Mitarbeiter: Sie bildet eine Gruppe von Mitarbeitern ab. Näheres findet sich im Abschnitt 5.3.1.

- Um die Mitarbeiter zu beschäftigen, bietet Object-VersPlan die Klasse Aufgabe. Von dieser abstrakten Klasse werden die für den Anwendungskontext spezialisierten Aufgaben abgeleitet.

- Benutzer ist die allgemeine Repräsentation all derjenigen, die mit Object-VersPlan arbeiten. Spielleiter haben Zugriff auf alle von ihnen erzeugten Spiele. Teilnehmer werden einem Unternehmen zugeordnet und haben damit Zugriff auf alle global freigegebenen Objekte und auf alle Objekte, die zu ‘ihrem’ Unternehmen gehören und für Spieler verfügbar sind.

- Objekte der Klasse Planspiel-Wurzel sind die Wurzel aller Teilehierarchien. Zu jedem Planspiel gehört genau ein Objekt dieser Klasse.

Im Abschnitt 5.3 werden einige dieser Modell-Klassen und deren Anwendung näher vorgestellt.

Verbindungen schaffen die Modellbasis für logische Abhängigkeiten zwischen Objekten:

- Exemplare der Klasse Verbindung werden für die Abbildung von Relationen benutzt: Eine solche Klasse verbindet beispielsweise Modell-Objekte mit View/Controller-Paaren und findet Verwendung für die Teilbeziehungen zwischen Bestandteilen der Teilehierarchie.

- Aktive Verbindungen modellieren Simulationsverbindungen. Näheres hierzu findet sich im Abschnitt 5.1.2.

231 Vgl. Felderer, B., Homburg, S. (Makroökonomik, 1991), S. 89 ff.

Die Subklassen von Netzwerkmodell fassen Netzwerkobjekte zu aggregierten Modellen zusammen und bieten Eigenschaften und Fähigkeiten zur deren Handhabung:

- Eine Teilehierarchie bildet den Rahmen für einen Ausschnitt aus globalen Teilehierarchie eines Planspiels.

- Die Klasse Ablaufmodell faßt Simulationsobjekte und deren Simulationsverbindungen zu Abbildungen der qualitativen Dynamik zusammen.

- Berichte enthalten Objekte zur Visualisierung von Simulationsobjekten.

- Ein Spiel besteht aus den Protokollen der Simulationen. Spiele werden von Spielleitern verwaltet und gehören eineindeutig zu einer Planspiel-Wurzel.

Neben diesen Modell-Klassen wurden noch eine ganze Reihe unterstützender Erweiterungen des Smalltalk-Laufzeitsystems um Hilfsklassen vorgenommen. Es handelt sich dabei beispielsweise um Klassen zur Behandlung von Funktionen, Zufallszahlen und deren Verteilungen sowie um Möglichkeiten der Speicherung und Visualierung von Zeitreihen.

### View-Klassen

Object-VersPlan benutzt zur Repräsentation im wesentlichen die vom System vorgebenen Klassen. Für die Darstellung der Objekte in den Teilehierarchien, in Ablaufmodellen und Berichten wurden spezielle View-Klassen implementiert.

*Tabelle 20: Klassenhierarchie der View-Klassen*

| Klassenbezeichnung |
|---|
| Netzwerk-Grafikobjekt |
| Ellipse |
| Ellipse mit Zusatzfeld |
| Text |
| Rechteck |
| Rechteck mit Zusatzfeld |
| eingebettetes Objekt |
| Verbindung |
| Geschäftsgrafik |
| Tortendiagramm |
| ... |

Die View-Klassen bieten sehr allgemeine Lösungen. Deshalb können auch solche Elemente in Form vorgefertigter Klassenbibliotheken zugekauft werden. Im vorliegenden

Beispiel sind dies die Klassen zur Visualisierung von Geschäftsgrafiken. Die Klassen zur Darstellung von Netzwerkobjekten sind einfach zu entwickeln und wurden daher selbst erstellt.

### Controller-Klassen

Object-VersPlan bietet im wesentlichen drei Gruppen von Controller-Klassen:

- Die wichtigsten Controller-Objekte sind die Editoren zum Bearbeiten der Netzwerkmodelle. Diese Werkzeuge werden im Abschnitt 5.2 detailliert vorgestellt.

- Dialoge ermöglichen den Zugriff auf die Eigenschaften sämtlicher Objekte.

- Es stehen darüberhinaus Controller für alle Netzwerkobjekte (Verbindungen und Simulationsobjekte) zur Verfügung.

Die folgende Tabelle zeigt, wie diese Klassen in drei Teilbäumen angeordnet werden.

*Tabelle 21: Klassenhierarchie der Controller-Klassen*

| Klassenbezeichnung |
|---|
| Editor |
| Start-Up |
| Teileeditor |
| Ablaufeditor |
| Berichtseditor |
| Inspektor |
| Simulator |
| Debugger |
| Entscheidungen-Editor |
| Dialog |
| Meldung |
| ... |
| Eigenschaftsdialog |
| ... |
| Netzwerkobjekt-Controller |
| Knoten |
| Berichtsobjekt |
| ... |
| Kante |

## 5.1.2. Simulation mit Object-VersPlan

Wie im vorangegangenen Abschnitt bereits angedeutet wurde, ist eine der grundlegenden Funktionen von Object-VersPlan die Möglichkeit zur Durchführung von Simulationen. Diese sind auf die Anforderungen von Unternehmensplanspielen zugeschnitten (vgl. 2.3).

### Annahmen

Das Simulationskonzept von Object-VersPlan basiert auf zwei Grundannahmen:

- „Tätigkeiten verbrauchen keine Zeit“ - die Dynamik des Modells spiegelt sich in der Entwicklung der Eigenschaften der Systembestandteile und der Veränderung ihrer Anordnung von Spielperiode zu Spielperiode wider.

- „Es werden Aggregate modelliert“ - die Modellierung konzentriert sich im wesentlichen auf makroökonomische Betrachtungen, eine streng mikroökonomisch fundierte Betrachtung individuellen Verhaltens findet nicht statt.

### Implementierung

Die Umsetzung unter Berücksichtigung der oben genannten Anforderungen findet sich in den Klassen zu den Modell-Objekten Simulationsobjekt und Spiel.[232]

- Die Systemelemente kommunizieren miteinander über Nachrichten, die Ergeignisse auslösen. Im Modell werden diese Kommunikationsbeziehungen durch Exemplare der Klasse aktive Simulationsverbindung abgebildet. Einige komplexere Objekte verfügen zudem über interne Simulationsverbindungen, die eine Art verdecktes Subsystem bilden.[233]

- Da es aus sachlichen Gründen oft wichtig ist, in welcher Reihenfolge einzelne Aktionen abgearbeitet werden, gibt es zwei Kontrollmöglichkeiten: Prioritäten und Bedingungen.

---

232 Dieser Mechanismus arbeitet ähnlich der Bedingung/Ereignis-Systeme in der Petrinetz-Theorie mit aktiven und passiven Elementen, mit Verbindungen zwischen den Systembestandteilen und mit Marken, mittels derer bedingte Abläufe modelliert werden.

233 Dieser Mechanismus wurde gewählt, um bei komplexeren, wiederverwendbaren Objekten die Kompliziertheit - ausgedrückt durch den Umfang des Protokolls und die Zahl der notwendigen Verknüpfungen - nicht unnötig ansteigen zu lassen. Ein erfreulicher Nebeneffekt ist die Tatsache, daß interne Simulationsverbindungen wesentlich schneller abgearbeitet werden. Der Nachteil liegt in der geringeren Flexibilität, da Erweiterungen der Funktionalität durch ‘echte’ Programmierung erfolgen müssen. Deshalb wurde dieser Ansatz vorzugsweise für sehr komplexe, universell anwendbare Objekte - wie etwa den Makler - gewählt.

– Ereignisse können parallel abgearbeitet werden. Da herkömmliche Prozessoren jedoch nur eine Aktion gleichzeitig ausführen können[234], werden die durch die Ereignisse angestoßenen Aktionen in Prozesse verpackt und von einer Prozeßverwaltung dem Prozessor je nach Priorität zur Verarbeitung weitergegeben. Object-VersPlan benutzt dazu die integrierte Prozeßverwaltung von **Smalltalk**.[235]

– Es gibt zwar eine logische Reihenfolge der Ereignisse, aber keinen Unterschied des Simulationszeitpunktes. Da Aktivitäten keine Zeit verbrauchen, finden alle Ereignisse zum selben Simulationszeitpunkt - z.B. am Geschäftsjahresende statt.

– Die Dynamik des Systems läßt sich als qualitativ dynamisches Verhalten auffassen. Zeitliche Dynamik entsteht erst durch die Betrachtung der Systemzustände von Simulationszeitpunkt zu Simulationszeitpunkt.

Ein Beispiel zeigt die Simulationsverbindungen des Objekts Auktionator (vgl. 5.3.2):

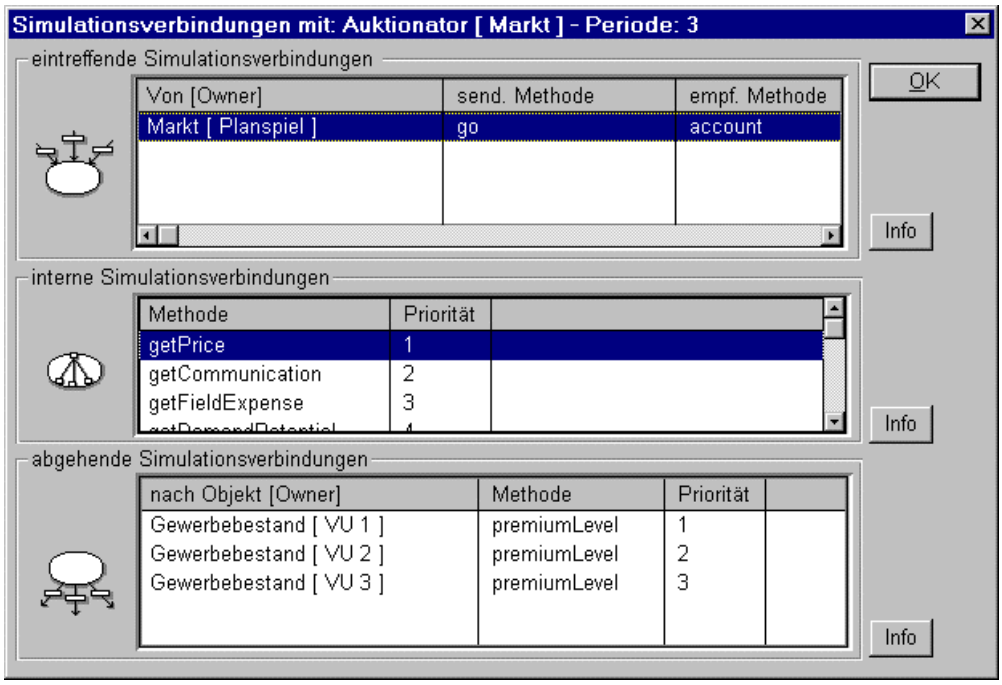



*Abbildung 13: Simulationsverbindungen*

234 Unter herkömmliche Porzessoren sollen auch die super-skalaren Mikroprosessoren gerechnet werden, die auf der Ebene des Maschinencodes mehrere Befehle gleichzeitig abarbeiten können. Aus Sicht der Anwendung oder des Betriebssystems arbeiten solche Prozessoren auch immer nur eine Aktion zur gleichen Zeit.

235 Diese Prozeßverwaltung darf nicht mit der Prozeßverwaltung moderner Multitasking-Betriebssysteme verwechselt werden. Aus Sicht des Betriebssystems arbeiten die Smalltalk-Prozesse in einem einzigen Prozeß.

Für die Verwaltung der Simulationsverbindungen steht dem Ersteller des Modells mit dem Abläufe-Editor ein mächtiges, graphisches Werkzeug zur Verfügung (vgl. 5.2)

Aus dem Blickwinkel der dahinterstehenden Datenbank ist eine Simulationsperiode eine lange Transaktion, während der die Objekte für andere Zugriffe gesperrt sind.[236] Bei Abbruch des Simulationsschrittes kann die Ausgangssituation, wie sie vor Beginn der Transaktion bestanden hat, wieder hergestellt werden.

### 5.1.3. Metamodell

Im Abschnitt 4.3 wurde versucht, die Problemdimensionen der Planspielentwicklung unabhängig von Einsatzzweck und dem abgebildeten Kontext zu erarbeiten. Die vorangegangenen beiden Abschnitte 5.1.1 und 5.1.2 haben die in Object-VersPlan verwendeten Ansätze zur Modellierung vorgestellt.

Folgt man den dort vorgeschlagenen Abstraktionen und Lösungen, so läßt sich das Simulationsmodells in einem Metamodell [237] zusammenfassen:

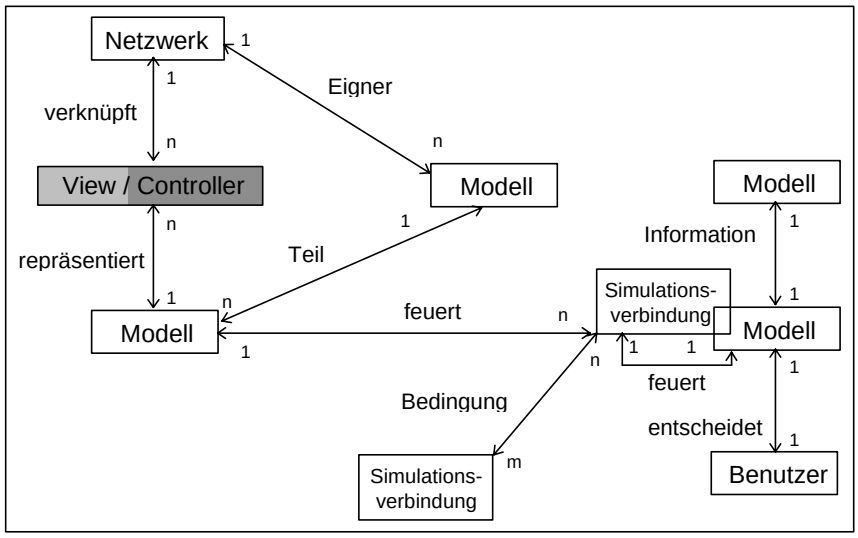



*Abbildung 14: Metamodell*

236 Diese Konstruktion folgt unmittelbar aus der Modellierung von Aggregaten und der Zeitpunktbetrachtung von Aktivitäten. Für dynamische Planspielseminare, bei denen die Zeit während des Seminargeschehens ständig weiterlaufen sollte, heißt das, daß es sich eigentlich nur um klassische Planspiele mit engeren Periodenabständen handelt. In der Praxis ist der Unterschied für die Teilnehmer allerdings nicht bedeutsam. Für die Planspielkonstruktion bedeutet dies aber einen sanfteren Übergang bei der Anpassung des Modells.

237 Zu Begriff und Anwendung von Metamodellen in der Softwareentwicklung:
Vgl. Hesse, W., Merbeth, G., Frölich, R. (Software-Entwicklung, 1992), S. 103 f.

Dieses Metamodell folgt im wesentlichen den im vierten Kapitel diskutierten Möglichkeiten zur Modellierung.

- Untergliederung der Modellierung nach M-V-C

- Getrennte Betrachtung der Aufbauorganiation (Struktur, statische Betrachtungsweise) und der Ablauforganisation (Dynamik)

### Implementierung

Mit den Mitteln objektorientierter Datenbanken läßt sich dieses Modell ohne semantische Lücke implementieren. Objekte (Klassen) müssen nicht unnötig segmentiert und können mitsamt ihres Verhaltens gespeichert werden. Ein objektorientiertes Datenbankmanagementsystem benötigt deshalb auch keine künstlichen Schlüsselattribute.

Für Object-VersPlan wurde mit ODBMS von VC eine reine Smalltalk-Datenbank gewählt, die schemafreie Speicherung und die direkte Verwendung von versionierten Verbindungen (links) erlaubt. Damit tritt die Datenbank in den Hintergrund und beschränkt sich auf die Realisierung von Persistenz, Zugriffskontrolle und Transaktionssteuerung.

## 5.2. Werkzeuge zur Entwicklung und Durchführung von Unternehmensplanspielen

Wesentlicher Bestandteil einer Plattform für Planspielentwicklung und -durchführung sind Werkzeuge, die den Benutzern problemorientierte Hilfen für die Bewältigung ihrer Aufgaben an die Hand geben sollen.

Die folgenden Abschnitte behandeln die Werkzeuge von Object-VersPlan.

1. Zuerst sollen die Zielgruppen und die Werkzeuge im Überblick vorgestellt werden.

2. Der folgende Abschnitt stellt die grundlegenden Konzepte für die Gestaltung der Werkzeuge dar.

3. Eine Vorstellung der Werkzeuge im Detail schließt die Betrachtungen ab.

### 5.2.1. Zielgruppen und Werkzeuge

Object-VersPlan ist angetreten, eine Entwicklungs- und Laufzeitplattform für Planspiele zu sein. Dementsprechend hat es für verschiedene Benutzergruppen - im folgenden mit Zielgruppen bezeichnet - und deren Aufgaben Unterstützung in Form von Werkzeugen anzubieten.

*Tabelle 22: Zielgruppen*

| Zielgruppe | |
|---|---|
| Konstrukteure | Konstruieren Planspiele |
| | Entwickeln wiederverwendbare Objekte |
| Spielleiter | Leiten Planspielseminare |
| Teilnehmer | spielen als Vorstand eines Spiel-Unternehmens |

Eine weitere Tabelle zeigt in einer Übersicht die Zielgruppen, die Aufgaben, für die sie Unterstützung benötigen und die Werkzeuge, die Object-VersPlan hierfür anbietet:

*Tabelle 23: Zielgruppen, Aufgaben und Werkzeuge*

| Zielgruppe | Benötigen Unterstützung für | Hilfe vom Werkzeug |
|---|---|---|
| Konstrukteure | Aufbau von Teilehierarchien | Teile-Editor |
| | Aufbau von Ablaufmodellen | Ablauf-Editor |
| | Test des Modelles und der Wirkzusammenhänge | Simulator, Debugger<br>Berichte-Editor |
| Spielleiter | Erstellung von Auswertungen für die Teilnehmer | Berichte-Editor |
| | Erstellung von Auswertungen für eigene Zwecke | Berichte-Editor |
| | Anpassung des Modells an individuelle Seminarsituation | Teile-Editor<br>Ablauf-Editor |
| | Eingabe der Entscheidungen | Entscheidungen-Manager |
| | Kontrolle der Simulation | Simulator, Debugger |
| | Verwaltung der Benutzer | Benutzer-Manager |
| Teilnehmer | Eingabe von Entscheidungen | Entscheidungen-Manager |
| | Ansicht von Berichten | |
| | Erstellung eigener Berichte | |

Die detaillierte Beschreibung der einzelnen Werkzeuge findet sich im Abschnitt 5.2.

## 5.2.2. Konzepte

Im folgenden sollen die Konzepte vorgestellt werden, denen die Entwicklung der Werkzeuge von Object-VersPlan folgt.

### Zielgruppenorientierung

Eine der maßgeblichen Forderungen anwendungsbestimmter Planspiele (vgl. 3.2) ist die Unterstützung der Benutzer mit angepaßten Werkzeugen, die ihren Bedürfnissen entsprechen.

Damit die Benutzer die Werkzeuge verstehen und gebrauchen können, auch wenn sie - wie Spielleiter und Teilnehmer - mit den technischen Aspekten von Softwaresystemen nicht vertraut sind, müssen vertraute Metaphern verwendet werden: *Budde* et al schlagen hierfür die Begriffe Werkzeuge und Materialien vor:[238]

- Editoren werden als Werkzeuge zum Bearbeiten von Materialien aufgefaßt. Ihre Gestaltung orientiert sich an den Tätigkeiten, die die Benutzer mit ihnen ausführen müssen. Für die benutzerorientierte Implementierung dieser Werkzeuge gibt *Constantine* einen simplen, aber wirkungsvollen Rat: „think like an user“[239]

- Materialien sind die Elemente, aus denen die Mikrowelt des Planspiels konstruiert wird. Für sie werden Begriffe gebraucht, die den Benutzern vertraut sind. Materialien der Planspielentwicklung finden sich in der Klassenbibliothek.

An dieser Stelle wird auf eine Diskussion über Aspekte der Softwareergonomie verzichtet. Es soll vielmehr das Potential des objektorientierten Ansatzes zur benutzerorientierten Gestaltung von Werkzeugen aufgezeigt werden.

### Paradigmenkonstanz

Materialien (aus der Klassenbibliothek) und Werkzeuge sind in starkem Maße voneinander abhängig - die Werkzeuge müssen das Protokoll und die Konzepte der Materialien benutzen, um mit ihnen kommunizieren und sie bearbeiten zu können. Deshalb ist selbstverständlich, daß auch bei der Implementierung der Werkzeuge die Paradigmen der Objektorientierung durchgehalten werden müssen.

Anhand der beiden Prinzipien M-V-C und Vererbung soll gezeigt werden, wie zwei Grundprinzipien der Objektorientierung für gutes Design der Werkzeuge einsetzt werden:

---

238 Vgl. Budde, R. et al (Anwendungssysteme, 1991), S. 191 ff.

239 Constantine, L. zitiert in: Thelen, B. (Ordnung, 1994) S. 81

## Model-View-Controller

Die Werkzeuge von Object-VersPlan können ganz im Sinne von M-V-C (vgl. 4.3.1) als View/Controller-Paare, die der Bearbeitung und Visualisierung von Modell-Objekten dienen, betrachtet werden:

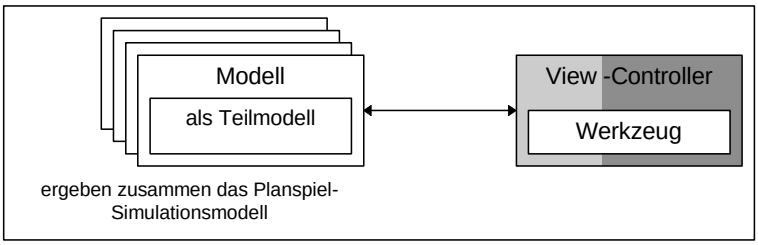



*Abbildung 15: Werkzeuge und M-V-C*

## Nutzung der Vererbung

Werkzeuge werden in einem eigenen Teilbaum der Controller-Klassenhierarchie repräsentiert. Durch die Nutzung der Möglichkeiten der Vererbung (vgl. 4.1.2) kann die Anpassung der Werkzeuge an die Zielgruppen mit verhältnismäßig geringem Aufwand realisiert werden:

– Es müssen immer nur diejenigen Funktionen implementiert werden, die die Superklasse in der Klassenhierarchie noch nicht besitzt oder die von denen der Superklasse abweichen.

– Für jede Anwendergruppe kann durch Vererbung ein geeignetes Werkzeug abgeleitet werden.

Das Beispiel der Teile-Editoren soll dies verdeutlichen:

*Abbildung 16: Ausschnitt aus der Klassenhierarchie für die Controller*

| Controller | |
|---|---|
| ... | |
| Editor | allgemeine Editor-Funktionen |
| Teile-Editor | Grapheneditor für Konstrukteure |
| Berichte-Editor | Grapheneditor für Spielleiter |
| Inspektor | Berichte-Editor für Spieler |
| ... | |

Editoren implementieren nur allgemeine Eigenschaften und Fähigkeiten - sie sorgen im wesentlichen für ein einheitliches Protokoll, eine einheitliche Benutzerschnittstelle und damit für eine einheitliche Bedienung.[240]

Teile-Editoren stellen alle notwendigen Funktionen zur Verfügung, die zur Bearbeitung von Netzwerkmodellen und seinen Bestandteilen, den Netzwerkobjekten, benötigt werden. Zusätzlich verfügen Teile-Editoren über Funktionen zur Bearbeitung von Teilehierarchien, wie sie Konstrukteure fordern. Diese Funktionen werden nicht vererbt.

Berichte-Editoren erben alle Funktionen zur Bearbeitung von Netzwerkmodellen und Netzwerkobjekten. Sie verfügen zusätzlich über Funktionen zur Bearbeitung von Berichten, wie sie Spielleiter benötigen.

Inspektoren erben Funktionen zur Bearbeitung von Berichten. Sie verfügen zusätzlich über Funktionen, die es Spielern ermöglichen, interaktiv und ohne detaillierte Kenntnis der Modellinterna Berichte zu erzeugen. Nicht geerbt werden diejenigen Funktionen, die zur Erstellung globaler Berichte und zur Vergabe von Rechten notwendig sind.

Durch zusätzliche Modularität der Editorfunktionalität können diese Werkzeuge noch feiner an die Benutzerbedurfnisse angepaßt werden. Ein Beispiel hierfür wären ladbare Assistenten und veränderbare Bedienerführung (Menüs und Schaltleisten).

### 5.2.3. Beschreibung der Werkzeuge

Object-VersPlan bietet einen Satz von Werkzeugen, die die Entwicklung und Durchführung von Planspielen unterstützen. Jeder dieser Editoren greift direkt auf das im Abschnitt 5.1.3 beschriebene **Metamodell** zurück, je nach Aufgabe bearbeiten sie einen Ausschnitt aus diesem Modell.

240 Die einheitliche Benutzerführung der Editoren orientiert sind an den de facto-Standards heutiger Standardapplikationen. Es kann an dieser Stelle nicht über Sinn oder Unsinn dieser Implementierungen diskutiert werden. Es hat sich eben - getrieben von Microsoft - ein de-facto-Standard etabliert. Die Mehrzahl der Computerbenutzer sind an die Bedienung dieser Applikationen gewöhnt und erwarten Analoges in allen anderen Programmen.
Vergleiche hierzu auch den Kommentar von *Müller-Zantop*: „Das Bessere hat nie gewonnen, sondern die Masse“. Müller-Zantop, S. (Megatrends, 1995), S. 84

## Der Editor für die Aufbauorganisation

Das grundlegende Werkzeug für die Erstellung von Planspielmodellen ist der Teile-Editor. Mit ihm erstellt der Konstrukteur die **Aufbauorganisation** eines Planspiels in Form einer Teilehierarchie:

- Alle Objekte, die am Planspiel-Simulationsmodell teilnehmen, werden mit dem Teile-Editor erzeugt. Objekte, die hier nicht erstellt wurden, können in der Simulation nicht verwendet werden.

- Damit die Modelle übersichtlicher sind, können Teilmodelle gebildet werden; diese werden durch Modell-Objekte vom Typ Teilehierarchie verwaltet. Die Gesamtheit der Teilmodelle ergibt die große Teilehierarchie eines Planspiels.

- Teilmodelle - wie beispielsweise ein Spielunternehmen und seine Aufbauorganisation - können als wiederverwendbare Bausteine in verschiedenen Planspielen Verwendung finden.[241]

- Es stehen die für graphisch orientierte Editoren üblichen Funktionen zur Bearbeitung der Objekteigenschaften, zum Kopieren, Ausschneiden und Einfügen, Drucken etc. zur Verfügung.

Auf der folgenden Grafik sieht man den vom Teile-Editor bearbeiteten Ausschnitt aus dem **Metamodell**:

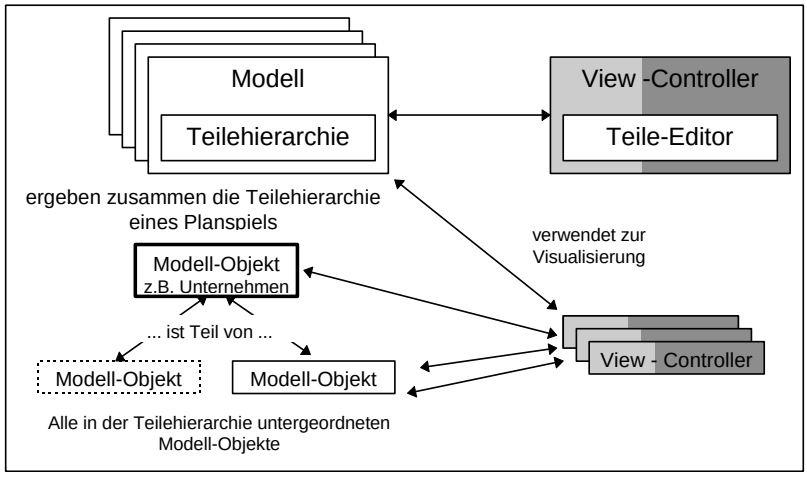



*Abbildung 17: Metamodell aus Sicht des Teile-Editors*

241 Mit Object-VersPlan ist es möglich, Unternehmen unterschiedlicher Organisationsformen in Bibliotheken abzulegen und bei Bedarf in einem Planspielmodell auf einem simulierten Markt konkurrieren zu lassen.
Die Verwendung von Teilmodellen in mehreren Planspielen ist allerdings nicht gleichzeitig möglich. Für jedes Planspiel sollte eine eigene Kopie verwendet werden.

Die folgende Abbildung zeigt den Teile-Editor mit einem kleinen Beispiel für eine Teilehierarchie eines Spielunternehmens:

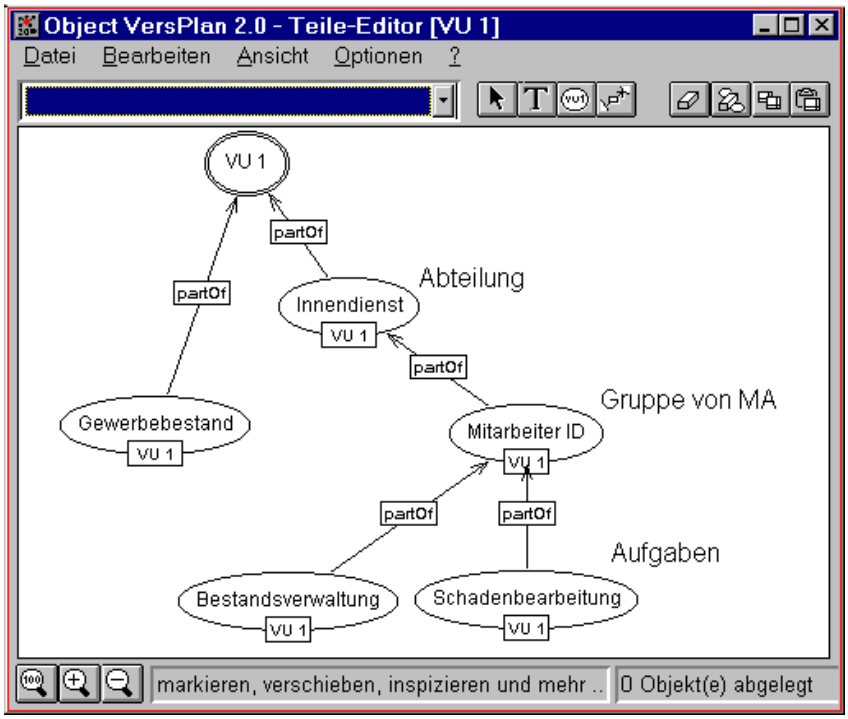



*Abbildung 18: Teile-Editor*

Weitere Beispiele für Ausschnitte aus der Teilehierarchie eines Planspiels finden sich im Abschnitt 5.3.1.

### Der Editor für die Ablauforganisation

Während der Teile-Editor für die Erzeugung und Anordnung der Objekte in einer Teilehierarchie zuständig ist, leistet ein Ablauf-Editor die Verknüpfung der Objekte miteinander: Es entstehen Abläufe, die die Dynamik des Modells beschreiben.

– Abläufe basieren auf den in Abschnitt 5.1.2 vorgestellten Konzepten: Ein Netz aus Kommunikationsbeziehungen (hier Simulationsverbindungen genannt) entsteht, indem über Nachrichten Daten ausgetauscht und Aktionen angestoßen werden.

– Jeder Ablauf ist einem Modell-Objekt zugeordnet. Dabei dürfen in diesem Ablaufmodell dürfen nur globale Objekte (beispielsweise der Spielleiter) und Objekte, die Elemente der Teilehierarchie dieses Modell-Objektes sind, verwendet werden.

– Jedem Objekt können beliebig viele Ablaufmodelle zugeordnet werden. Damit ist die Darstellung kleinerer und übersichtlicherer Modelle möglich.

– Der Ablauf-Editor erbt die allgemeinen Funktionen zur Objektbearbeitung vom Teile-Editor und bietet zudem Möglichkeiten zur Bearbeitung der Simulationsverbindungen.

Die Abbildungen zeigen den Ausschnitt aus dem **Metamodell**, der mit Hilfe des Ablauf-Editors bearbeitet wird und ein Beispiel für den Ablauf einer Simulationsperiode: [242]

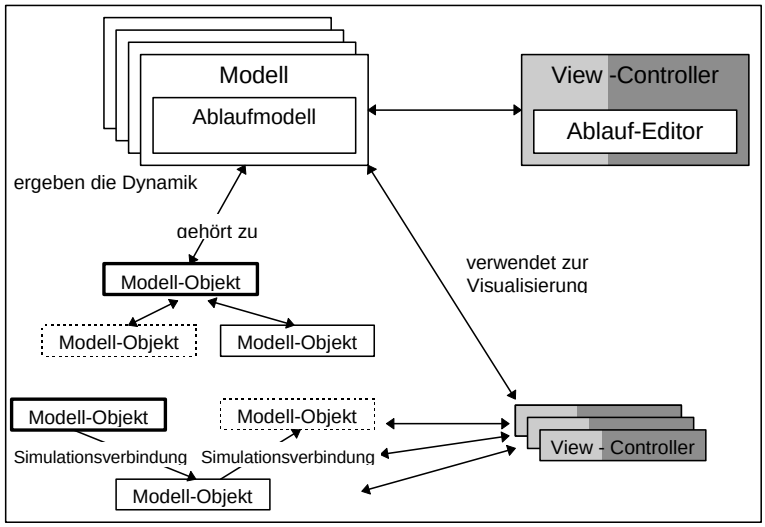
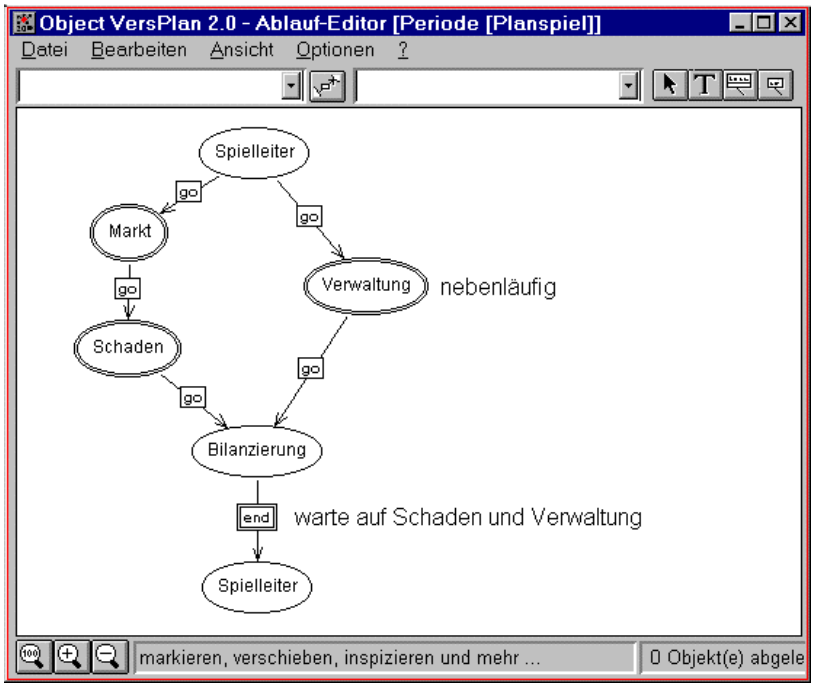



*Abbildung 19: Metamodell aus Sicht des Ablauf-Editors*




242 Die Beschreibung dieses Beispiels findet sich in Abschnitt 5.3.2.

*Abbildung 20: Ablauf-Editor*

Beispiele für Abläufe finden sich im Abschnitt 5.3.2.

### Entscheidungen-Manager

Wesentlicher Aspekt eines Unternehmensplanspiels ist die Möglichkeit der Spieler, durch Entscheidungen[243] mit ihren Unternehmen direkten Einfluß auf das simulierte Geschehen zu nehmen. Object-VersPlan bietet mit dem Entscheidungen-Manager ein Werkzeug an, das, abhängig vom Status des Benutzers (sei er Spieler, Spielleiter oder Konstrukteur), alle verfügbaren Entscheidungen auflistet:

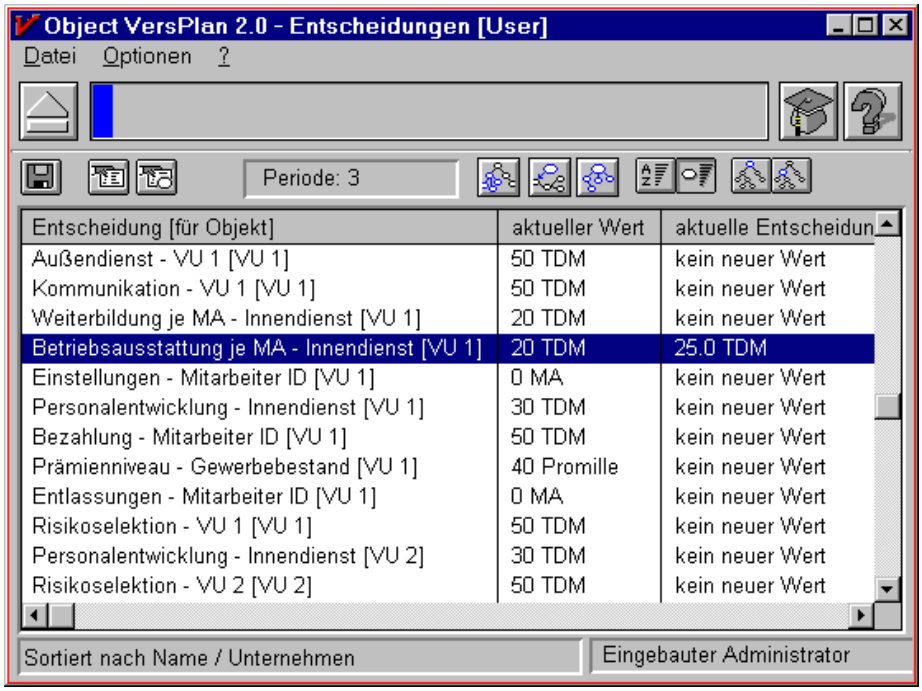

Object VersPlan 2.0 - Entscheidungen [User]
Datei Optionen ?
Periode: 3
Entscheidung [für Objekt]
aktueller Wert
aktuelle Entscheidun
Außendienst - VU 1 [VU 1] 50 TDM kein neuer Wert
Kommunikation - VU 1 [VU 1] 50 TDM kein neuer Wert
Weiterbildung je MA - Innendienst [VU 1] 20 TDM kein neuer Wert
Betriebsausstattung je MA - Innendienst [VU 1] 20 TDM 25.0 TDM
Einstellungen - Mitarbeiter ID [VU 1] 0 MA kein neuer Wert
Personalentwicklung - Innendienst [VU 1] 30 TDM kein neuer Wert
Bezahlung - Mitarbeiter ID [VU 1] 50 TDM kein neuer Wert
Prämienniveau - Gewerbebestand [VU 1] 40 Promille kein neuer Wert
Entlassungen - Mitarbeiter ID [VU 1] 0 MA kein neuer Wert
Risikoselektion - VU 1 [VU 1] 50 TDM kein neuer Wert
Personalentwicklung - Innendienst [VU 2] 30 TDM kein neuer Wert
Risikoselektion - VU 2 [VU 2] 50 TDM kein neuer Wert
Sortiert nach Name / Unternehmen
Eingebauter Administrator

*Abbildung 21: Entscheidungen-Manager*

Für Spielleiter und Konstrukteure gibt es weiterhin die Möglichkeit, zur Entscheidungsfindung alle anderen Editoren zu benutzen. Sie erhalten dann eine Auflistung aller möglichen Entscheidungsmöglichkeiten bezüglich eines speziellen Objekts.

Damit der Entscheidende einen Überblick über bisher getroffene Entscheidungen erhält, bietet der Eingabedialog eine graphische Darstellung der Zeitreihe an:

---

243 Wenn in der Folge von ‘Entscheidungen’ die Rede ist, so sind damit immer die Eingriffsmöglichkeiten der Teilnehmer gemeint. Spielleiter und Konstrukteure können darüberhinaus Entscheidungen über Modellstruktur und über das Verhalten der Systemelemente treffen.

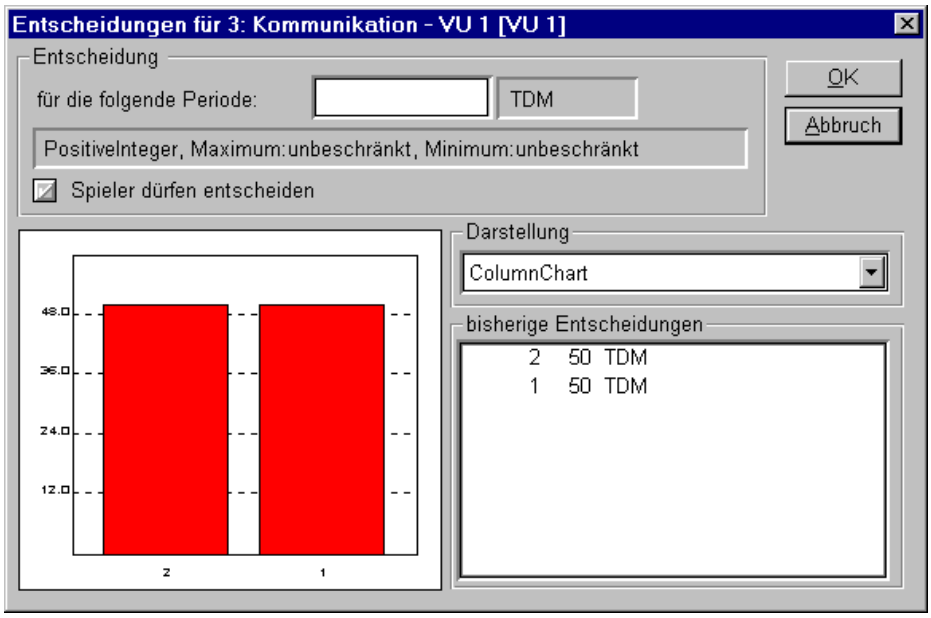



*Abbildung 22: Eingabedialog für Entscheidungen*

## Simulatorwerkzeuge

Das Konzept der Simulation wurde bereits im Abschnitt 5.1.2 behandelt. Nun sollen die Werkzeuge vorgestellt werden, die zur Durchführung und Kontrolle der Simulation zur Verfügung stehen: Es sind dies der Simulator und der Debugger, ein zusätzliches Hilfsmittel für die Verwaltung und Analyse von Fehlermeldungen. Sie sind die beiden Hilfen für Spielleitung und Konstrukteur, die die Kontrolle über die Simulation in der Entwicklungsphase, während der Kalibrierung und bei der Durchführung des Planspiels ermöglichen.

Mit dem Simulator wird versucht, dem bereits diskutierten Problem der Eigenständigkeit der an der Simulation beteiligten Objekte Rechnung zu tragen. Man kann sich den Simulator als die zentrale Instanz vorstellen, die die Oberaufsicht über die Kommunikationsströme während des Simulationslaufes hält.

Die Aufgaben der Simulatorwerkzeuge in zwei Punkten zusammengefaßt:

1. Augenscheinlichste Aufgabe ist die Protokollierung des Simulationsablaufs zur Überwachung während der Phasen Entwicklung, Validierung und Durchführung. Ein Fehlerprotokoll - verwaltet vom Debugger - listet auf Wunsch die während der Laufzeit aufgetretenen Warnungen und Fehlermeldungen auf und bietet Möglichkeiten zur Analyse möglicher Fehlerursachen.

2. Eher im Hintergrund laufen Kontrolle und Verwaltung der Prozesse der Simulation, damit ein Abbrechen und Unterbrechen der Simulation jederzeit möglich ist.

Selbstverständlich folgt auch die Implementierung der Simulator-Werkzeuge dem Ebenenmodell nach M-V-C:

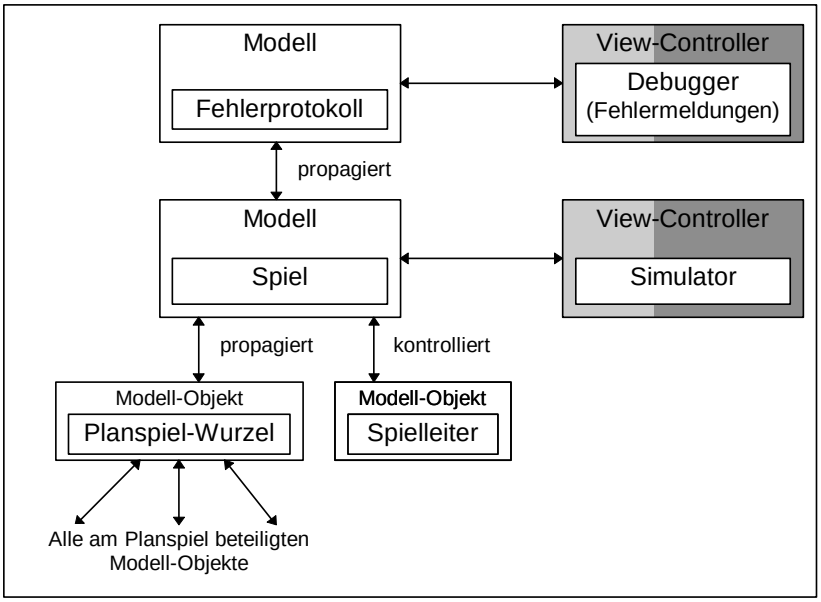



*Abbildung 23: Metamodell aus Sicht der Simulatorwerkzeuge*

Das Modell-Objekt für den Simulator ist das Spiel, das als Spezialisierung der abstraken Klasse Netzwerkmodell in die Klassenhierarchie eingehängt wird.

Jedem Spiel ist genau ein Objekt der Klasse Planspiel-Wurzel zugeordnet. Über dieses Objekt wird die Vorbereitung der Simulationsschritte propagiert[244]. Die Planspiel-Wurzel ist so etwas wie die zentrale Instanz, d.h., alle Elemente des Simulationsmodells haben - direkt oder über mehrere Ebenen - eine Teile-Beziehung zu ihr.

Ein weiteres Objekt spielt eine Sonderrolle: Der Spielleiter. Mit dem Spielleiter, der Instanz der Klasse Spielleiter ist, wird der externe Einfluß auf das Simulationsmodell abgebildet. Ein Spielleiter kontrolliert die Simulation. Er kann Simulationen starten, unterbrechen und beenden. Spielleiter bringen aber auch externe Effekte wie zum Beispiel die Einflüsse der allgemeinen Wirtschaftsentwicklung oder grundsätzliche Veränderungen in der Riskostruktur, hervorgerufen durch unvorhersehbare Ereignisse und Strukturbrüche, in das Modell ein. Einem Spiel ist genau ein Spielleiter zugeordnet.

Die folgende Abbildung zeigt die Aufgabenteilung während der Abarbeitung einer Simulationsperiode:

244 Zur Vorbereitung einer Simulationsperiode gehört das Zurücksetzen der Semaphore in den Simulationsverbindungen und die Vorbereitung der Werte aller an der Simulation beteiligten Objekte.

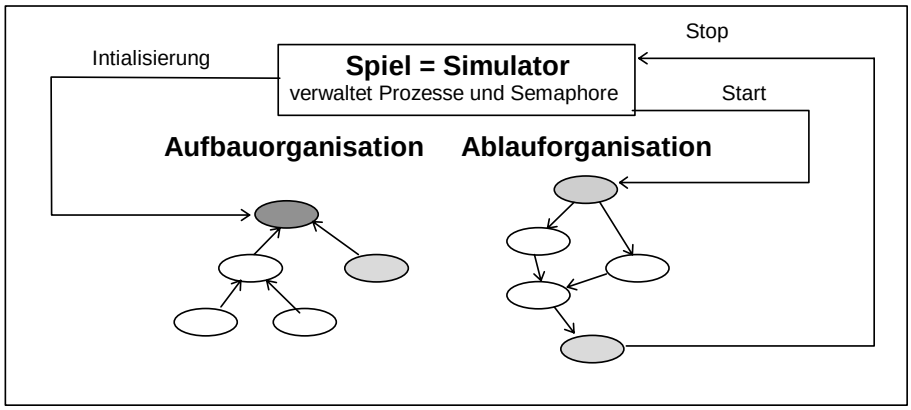



*Abbildung 24: Aufgabenteilung während der Simulation*

Wie bereits im Abschnitt 5.1.2 vorgestellt wurde, ermöglicht Object-VersPlan die Abarbeitung nebenläufiger Prozesse: Aufgaben können parallel ausgeführt werden und über Bedigungen und Ereignisse in eine logische Reihenfolge gebracht werden. Die Aufgabe des Simulators ist es, diese Prozesse zu kontrollieren.[245]

Abbildung 23 zeigt, daß Exemplare der View/Controller-Klassen Simulator und Debugger für die Schnittstelle zum Benutzer verantwortlich sind.

Simulator und den Debugger finden sich in den Abbildungen 25 und 26:

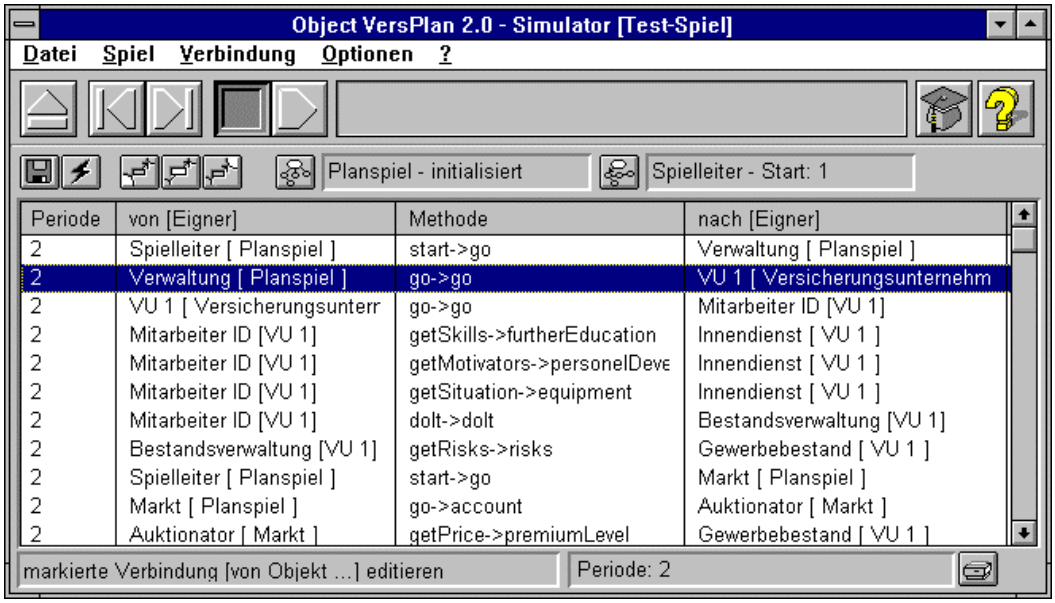

Object VersPlan 2.0 - Simulator [Test-Spiel]
Datei Spiel Verbindung Optionen ?
Planspiel - initialisiert
Spielleiter - Start: 1
Periode | von [Eigner] | Methode | nach [Eigner]
2 | Spielleiter [ Planspiel ] | start->go | Verwaltung [ Planspiel ]
2 | Verwaltung [ Planspiel ] | go->go | VU 1 [ Versicherungsunternehm
2 | VU 1 [ Versicherungsunterr | go->go | Mitarbeiter ID [VU 1]
2 | Mitarbeiter ID [VU 1] | getSkills->furtherEducation | Innendienst [ VU 1 ]
2 | Mitarbeiter ID [VU 1] | getMotivators->personelDeve | Innendienst [ VU 1 ]
2 | Mitarbeiter ID [VU 1] | getSituation->equipment | Innendienst [ VU 1 ]
2 | Mitarbeiter ID [VU 1] | dolt->dolt | Bestandsverwaltung [VU 1]
2 | Bestandsverwaltung [VU 1] | getRisks->risks | Gewerbebestand [ VU 1 ]
2 | Spielleiter [ Planspiel ] | start->go | Markt [ Planspiel ]
2 | Markt [ Planspiel ] | go->account | Auktionator [ Markt ]
2 | Auktionator [ Markt ] | getPrice->premiumLevel | Gewerbebestand [ VU 1 ]
markierte Verbindung [von Objekt ...] editieren
Periode: 2

*Abbildung 25: Simulator*

245 Der Simulator bedient sich dazu der Prozeßverwaltung des Smalltalk-Systems und der Möglichkeiten der Transaktionsverwaltung des Datenbanksystems.

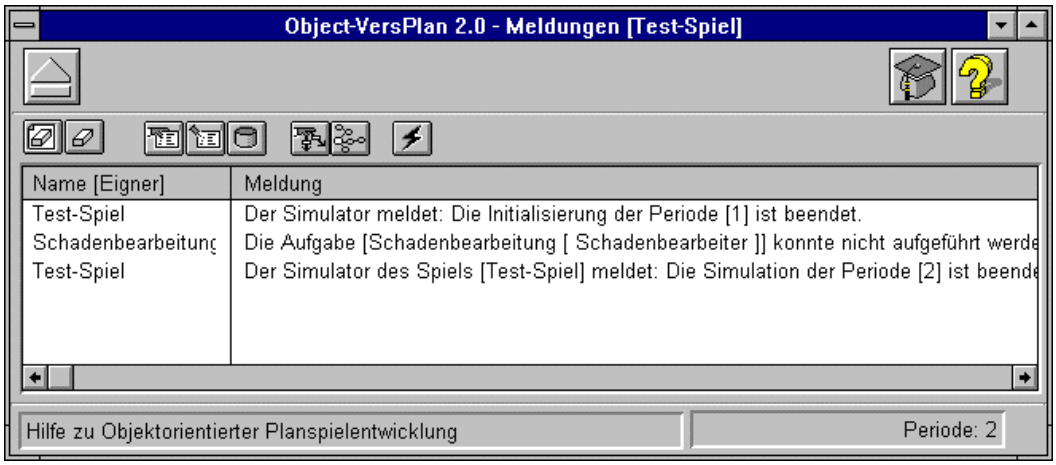

Object-VersPlan 2.0 - Meldungen [Test-Spiel]
Name [Eigner]
Meldung
Test-Spiel
Der Simulator meldet: Die Initialisierung der Periode [1] ist beendet.
Schadenbearbeitung
Die Aufgabe [Schadenbearbeitung [ Schadenbearbeiter ]] konnte nicht aufgeführt werde
Test-Spiel
Der Simulator des Spiels [Test-Spiel] meldet: Die Simulation der Periode [2] ist beende
Hilfe zu Objektorientierter Planspielentwicklung
Periode: 2

*Abbildung 26: Debugger*

### Berichte-Editor

Eine herausragende Stärke einer objektorientierten, nach dem M-V-C Ansatz implementierten, Applikation ist es, individuelle Ansichten des selben Modells für die Bedürfnisse des jeweiligen Benutzers anzupassen: Spieler erhalten Informationen zu ihrem Unternehmen und Zugang zu allgemein verfügbaren Daten wie beispielsweise Marktinformationen oder Geschäftsberichten der Konkurrenzunternehmen. Spielleiter können sich Übersichten zur Überwachung des Spielverlaufs zusammenstellen und mitverfolgen, welche Auswirkungen konkurrierende Entscheidungen der Spielunternehmen auf das Marktgeschehen haben. Für den Konstrukteur ist es wichtig, daß er bei der Kalibrierung der Wirkzusammenhänge wirksam unterstützt wird. Dazu kann er beispielsweise relevante Wirkfunktionen und Einflußgrößen in Berichten nebeneinander positionieren und während des Tests beobachten. Und immer gilt: Das Weglassen nicht relevanter Informationen, die Verwendung graphischer Visualisierungen zur Tendenzanzeige und der Rückgriff auf tabellarische Darstellungen (falls Detailinformationen im Vordergrund stehen), sind möglich.

Der Berichte-Editor ist das zentrale Werkzeug für diese Aufgaben:

- Jeder Bericht ist einem Objekt (beispielsweise einem Unternehmen) zugeordnet.
- Er kann zur Darstellung von Eigenschaften dieses Objektes und seiner in der Teilehierarchie untergeordneten Objekte verwendet werden.
- Zur Visualisierung kommen prinzipiell alle Unterklassen von View[246] zusammen mit dem Controller Berichtsobjekt[247] zum Einsatz.

246 vgl. 5.1.1: View-Klassen

– Berichte können als öffentlich oder privat deklariert werden: Entweder nur der Eigner (also etwa einem Unternehmen) oder alle nehmen Einsicht. Dies ist für dynamische Planspiele wichtig, bei denen die Teilnehmer selbst Zugriff auf das Modell haben.

– Durch den Abhängigkeitsmechanismus sind alle Berichte dynamisch: Sie passen sich automatisch den Veränderungen des Modells an.

– Selbstverständlich erbt der Berichte-Editor die allgemeinen Funktionen vom Teile-Editor und ermöglicht neben dem interaktiven Arbeiten direkt am Rechner[248] auch die Ausgabe auf Papier, wie sie für klassische Planspielseminare benötigt wird.

Auch der Berichte-Editor folgt dem Ebenenmodell nach M-V-C:

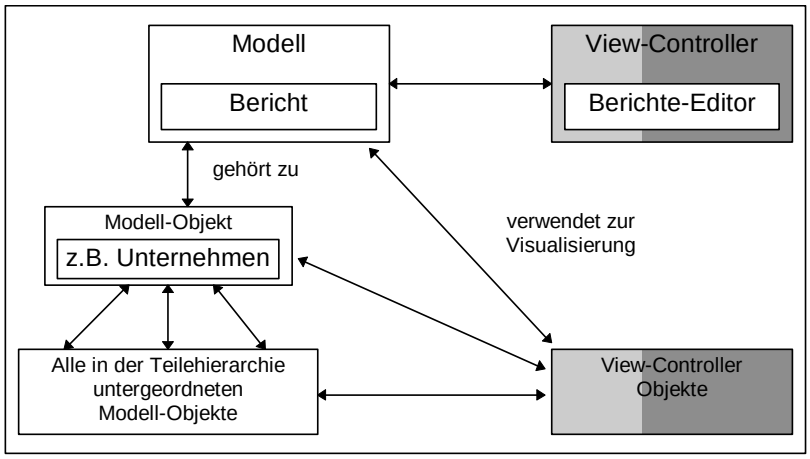



*Abbildung 27: Metamodell aus Sicht des Berichte-Editors*

Das folgende Beispiel zeigt einen Bericht für einen Konstrukteur, der die korrekte Wirkung der Aufwände für die Weiterbildung testen möchte:

247 vgl. 5.1.1: Controller-Klassen

248 Wie alle für Spielleiter und Konstruktuer geeigneten Editoren, so können auch vom Berichte-Editor alle Informationen über das zugrundeliegende Modell erreicht werden: Berichtsobjekte sind lediglich eine Möglichkeit für View/Controller-Paare, die über den Abhängigkeitsmechanismus mit dem Modell verküpft sind. Verfolgt man diese Verknüpfung, so kann man sich schrittweise die gewünschten Informationen erarbeiten.

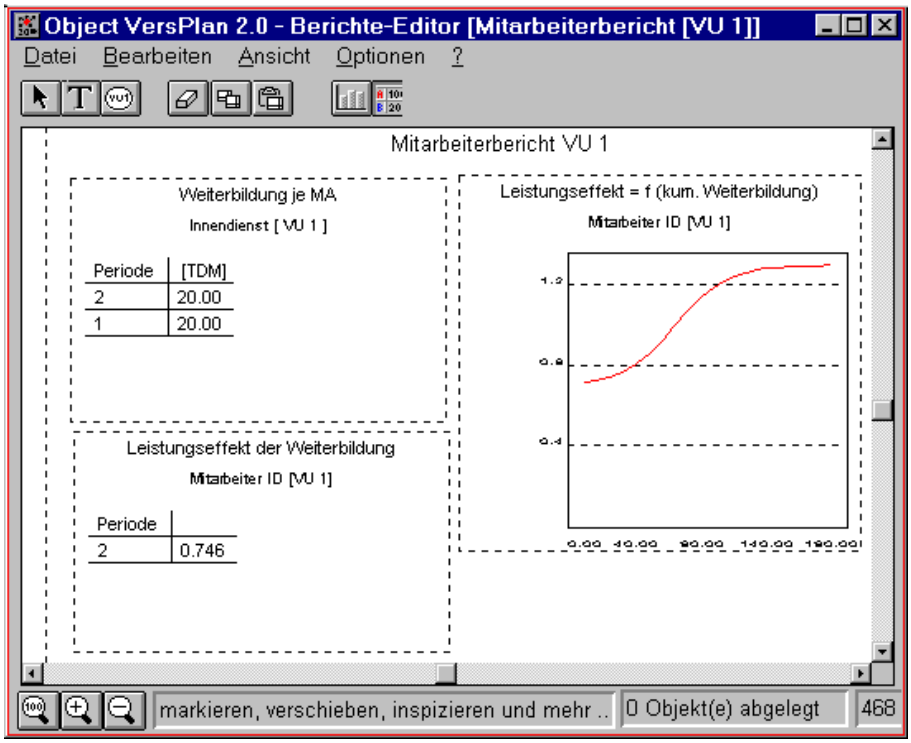



*Abbildung 28: Bericht*

## Inspektor

Wenn die Seminarsituation (vgl. 3.2.1) vorsieht, daß Teilnehmer interaktiv am Rechner mit dem Modell arbeiten, ist ein Werkzeug hierfür notwendig.

Für diese Aufgabe ist der Inspektor gedacht. Bei diesem Werkzeug handelt es sich um eine Spezialisierung des Berichte-Editor, der nur zum Anzeigen - jedoch nicht zum Bearbeiten - von Berichten geeignet ist.

Spätere Versionen von Object-VersPlan werden erweiterte Möglichkeiten zulassen: Assistenten unterstützen den Benutzer bei der interaktiven Erstellung von Auswertungen. Spieler könnten dann selbst diejenigen Berichte aus den ihnen zugänglichen Informationen generieren, die Sie für Ihre spezifische Entscheidungssituation benötigen.

## Benutzer-Editor

Ebenso wie der Inspektor so ist auch der Benutzer-Editor für Seminarsituationen vorgesehen, in denen die Teilnehmer am Rechner arbeiten können. Dazu wird für jedes Team mindestens ein Benutzer erzeugt, der mit Rechten eines Spieler ausgestattet wird:

– Ein Spieler hat typischerweise Lesezugriff auf alle global verfügbaren Informationen (Berichte).

– Er ist einem Unternehmen zugeordnet: Dadurch kann der Spieler alle (vom Spielleiter freigegebenen) Entscheidungen selbst eingeben und Berichte zu seinem eigenen Unternehmen einsehen.

– Informationen über andere Unternehmen und interne Modellinformation, die der Spielleiter oder der Konstrukteur benötigen, sind für die Spieler nicht sichtbar.

### Datenbank-Manager

Ein letztes Werkzeug sei nur der Vollständigkeit halber aufgeführt: Der Datenbank-Manager ist zur Systemadministration für Konstrukteure und Spielleiter gedacht. Da alle Aktionen auf eine objektorientierte Datenbank abgewickelt werden, muß es auch eine Schnittstelle zu dieser Datenbank geben. Dementsprechend bietet der Datenbank-Manager Funktionen zur Kompression, Sicherung, Replikation und Restaurierung.

## .2.4. Systemarchitektur

Die in den vorangegangenen Abschnitten vorgestellten Werkzeuge wurden den Grundsätzen der Client-Server-Architektur[249] folgend in einem Entwicklungs- und Laufzeitsystem integriert. Die folgende Abbildung gibt einen Überblick:

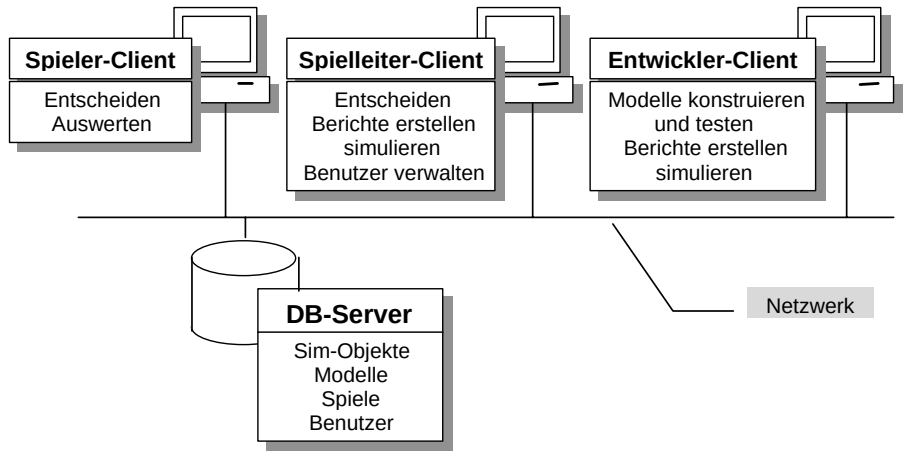



*Abbildung 29: Systemarchitektur von Object-VersPlan*

249 Es werden die englischen Begriffe client und server gebraucht, weil sie sich in der Fachsprache eingebürgert haben und eine eindeutige und gebräuchliche Übersetzung nicht verfügbar ist. Zu Client/Server-Systemen in der Praxis: vgl. Letters, F. (Client/Server, 1995), S. 36 ff.

### Spieler-Client

Die Architektur von Object-VersPlan läßt die Möglichkeit offen, daß die Teilnehmer (Spieler) selbst mit dem Modell agieren können: Sie können sich mit dem Inspektor Informationen über den aktuellen Zustand des Modells (Spielstand) einholen und die Entscheidungen selbst im Modell eingeben. Dafür ist ein Programm, der Spieler-Client, verfügbar.

### Spielleiter-Client

Für den Spielleiter stehen weitergehende Eingriffsmöglichkeiten in das Modell zur Verfügung: Zur Eingabe von Entscheidungen, der Erstellung und Anzeige von Berichten und Auswertungen, der Durchführung und Überwachung der Simulation und zur Verwaltung von Benutzern (Teilnehmer, Spieler) und Datenbank sind im Spielleiter-Client der Entscheidungen-Editor, der Berichte-Editor, die Simulatorwerkzeuge und der Datenbank-Manager verfügbar. Die Funktionalität des Spieler-Clients ist im Spielleiter-Clienten enthalten.

### Entwickler-Client

Die umfangreichste Unterstützung wird im Entwickler-Client-Programm dem Konstrukteur von Planspielmodellen angeboten. Er enthält zusätzlich zu den Werkzeugen des Spielleiter-Clients Editoren zum Bearbeiten von Teilehierarchien und Ablaufmodellen: Teile-Editor und Ablauf-Editor verfügen über umfangreiche Möglichkeiten zum Konstruieren und Testen von Planspielmodellen.

### Datenbank-Server

Alle am Planspiel beteiligten Objekte (Modelle und Berichte, Systembestandteile, Teilnehmer) werden in einer zentralen Objektdatenbank gespeichert. Diese Datenbank sichert durch einen Transaktionsmechanismus die Konsistenz bei komplexen Aktionen und sie koordiniert durch Schutzmechanismen die Zugriffe mehrerer gleichzeitiger Benutzer (beispielsweise der Spieler und der Spielleitung)[250].

250 In der vorliegenden Version von Object-VersPlan sind echter Client-Server-Betrieb und volle Nutzung der Transaktions- und Schutzmechanismen noch nicht implementiert. Dies liegt zum einen am Prototyp-Stadium vom Object-VersPlan, zum anderen an der fehlenden Verfügbarkeit eines ausgereiften und performanten Object-Servers für schemafreie Speicherung.

## 5.3. Die Anwendung von Object-VersPlan - dargestellt an ausgewählten Beispielen

### 5.3.1. Aufbauorganisation

Object-VersPlan trennt bei der Modellierung die statische Betrachtungsweise der Aufbauorganiation von der Abbildung der dynamischen Zusammenhänge in Form der Ablauforganisation. Der folgende Abschnitt zeigt an ausgewählten Beispielen Möglichkeiten zur Gestaltung der **Aufbauorganisation**.

#### Planspiel

Wie bereits im zweiten Kapitel ausgeführt wurde, wird das Planspiel in seiner Ganzheit als System verstanden. Dann ist es nur konsequent, die Abbildung der Aufbauorganisation im Simulationsmodell auch beim Objekt Planspiel beginnen zu lassen.

Die folgende Abbildung zeigt den oberen Teil einer möglichen Aufbauorganisation eines Versicherungsplanspiels:

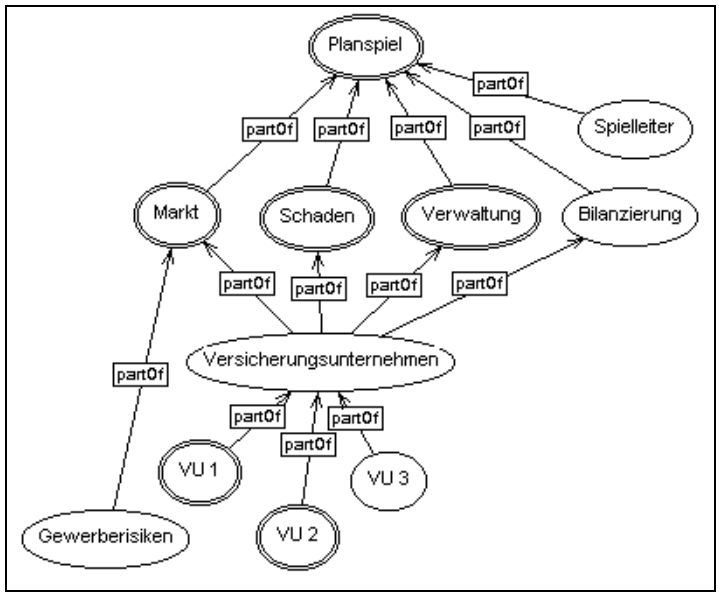



*Abbildung 30: Aufbauorganisation eines Planspiels*

Das Beispiel zeigt die Spitze der Aufbauorganisation eines kleinen Systemmodells für ein Versicherungsplanspiel: Drei Versicherungsunternehmen (VU 1, VU 2 und VU 3) konkurrieren mit einem Produkt (der Deckung von Gewerberisiken) auf einem Markt. Die Modellierung umfaßt die Simulation des Marktgeschehens, der Schadenfälle und der

Verwaltungstätigkeiten im Innendienst der Unternehmungen von der Antragsbearbeitung bis zur Bilanzierung. Dieses Beispielmodell wurde so einfach wie möglich gehalten, um die Grundideen der Modellierung durch Anwendung der im Verlauf der vorliegenden Arbeit beschriebenen objektorientierten Vorgehensweise klar aufzeigen zu können.

Dieses Planspielmodell gliedert sich im wesentlichen in zwei Bereiche:

1. Die Modellierung des Versicherungsmarktes, in dem die Spielunternehmen im Wettbewerb agieren.

2. Die Abbildung der Externalitäten, der exogenen Einflüsse auf das Modell - beispielsweise volkswirtschaftliche Einflüsse und sonstige Rahmenbedingungen -wie sie hier durch das Objekt des Spielleiters repräsentiert werden.

### Unternehmen

Die folgende Abbildung zeigt ein Beispiel für die Abbildung der Organisation eines Unternehmens mit nur wenigen Elementen:

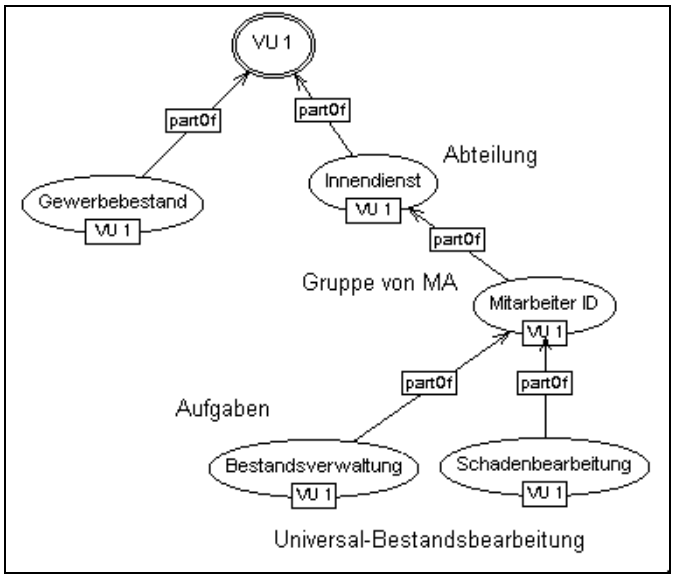



*Abbildung 31: Unternehmen*

Bei einer derart vereinfachten Modellierung werden die weiteren für die Funktionsfähigkeit des Modells wichtigen Aspekte wie beispielsweise der Außendienst oder die Anstrengungen für Kommunikation (beispielsweise Werbung) nur durch einen Wert

repräsentiert. Dieser Wert kann beispielsweise als Entscheidung zum Unternehmen modelliert werden.[251]

Das Beispiel zeigt die globalen Entscheidungen zum Objekt VU 1:

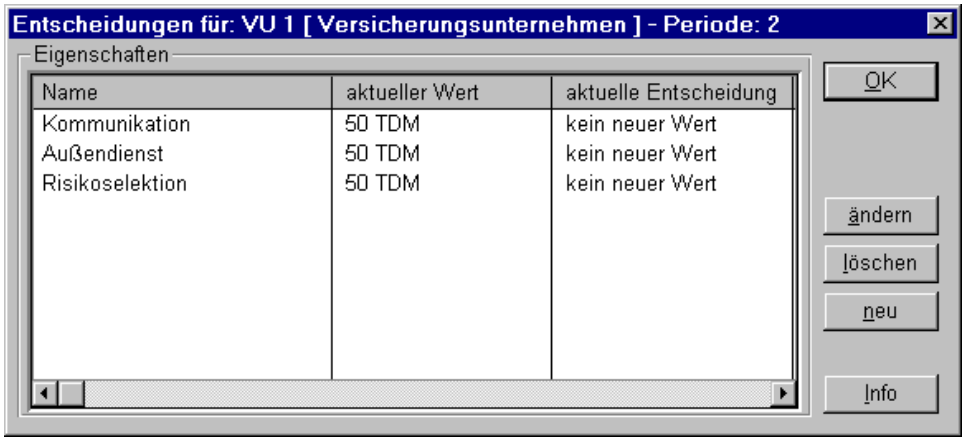



*Abbildung 32: Globale Entscheidungen zum Unternehmen*

Wie bei allen Entscheidungen, so steht auch hier dem Entscheidenden - hier also dem Teilnehmer - die Statistik der bisher getroffenen Entscheidungen zur Verfügung (vgl. 5.2.3).

### Risiken

Der für ein Versicherungsplanspiel zentrale Bereich ist das Produkt Versicherungsschutz. Object-VersPlan modelliert Versicherungsschutz dem Makro-Ansatz folgend als ein Kollektiv von versicherbaren Risiken[252]: Risiken, die im Rahmen der Vertragsbedingungen bei einem Versicherungsunternehmen Deckung finden, im Bestand (Portefeuille) geführt werden oder auch (noch) nicht versichert sind.

Diese allgemeine Art der Modellierung hat wesentliche Vorteile:

– Mit dem Objekt Risiko kann eine abstrakte Klasse entwickelt werden, die die Basiseigenschaften und -fähigkeiten eines Kollektivs besitzt: Es kann das Risikoprofil

251 Object-VersPlan bietet zu jedem Objekt die Möglichkeit, Eigenschaften in Form einfacher Werte hinzuzufügen. Diese Werte können im weiteren Verlauf der Simulation, also etwa der Marktsimulation, ausgewertet werden. Hiermit steht eine einfache Möglichkeit zur Verfügung, globale, hochaggregierte Entscheidungsmöglichkeiten in ein Modell aufzunehmen.

252 Zum Begriff der Versicherbarkeit von Risiken: vgl. Karten, W. (Versicherbarkeit, 1972), S. 279 ff. Wenn in der Folge von Risiken die Rede ist, so sind stets versicherbare Risiken gemeint. Der Begriff des Risikos wird in der Betriebswirtschaftlehre in vielen Varianten gebraucht. Eine Diskussion findet sich in: Helten, E. (Risiko, 1994), S. 19 ff.; vgl. a. Helten, E., Karten, W. (Kalkulation, 1983), S. 3 ff.

(ausgedrückt in den Schadenzahl- und Schadensummenverteilungen[253]) für die Menge relativ homogener Risiken abgebildet werden. Das Objekt hat die Fähigkeit, Schäden zu simulieren, und es besitzt elementare Vetragseigenschaften - wie etwa die durchschnittliche Versicherungssumme.

– Wenn versicherte und unversicherte Risiken analog abgebildet werden, kann der Prozeß der Veränderung der Risikoprofile sehr transparent und realitätsnah modelliert werden: Die Versicherungsunternehmen nehmen neue Risiken aus dem Bestand unversicherter Risiken durch Akquisition auf (Neugeschäft), während bereits versicherte Risiken durch Kündigung in das Kollektiv unversicherter Risiken zurückgehen (Storno, Kündigung, Vertragsablauf)[254]. Dabei kann berücksichtigt werden, daß diese Risiken unterschiedliche Risikoprofile aufweisen (‘Relativ schlechtes Geschäft wird abgegeben, während gutes Geschäft neu akquiriert wird und im Bestand gehalten werden soll’). Damit sich die Risikoprofile versicherter und nicht versicherter Risiken auch tatsächlich verändern, müssen die Wahrscheinlichkeitsverteilungen (Schadenzahl und Schadenhöhe oder Gesamtschaden) durch Faltung[255] neu berechnet werden.

  All diese Veränderungen können durch Maßnahmen der Versicherungsunternehmen beeinflußt werden. Im Abschnitt zur Marktsimulation werden die eben genannten Aspekte noch einmal aufgegriffen.

– Vor der abstrakten Klasse Risiko können bei Bedarf (unter Nutzung der Wiederverwendung) spezialisierte Formen von Risikokollektiven - also etwa Industrierisiken oder Gewerberisiken - abgeleitet werden.

  Für solch spezialisierten Risiken ist die Festlegung individueller Vertragsbedingungen (beispielsweise Selbstbehalt, maximale Deckungssummen oder Laufzeit) und spezifischer Möglichkeiten zur Abbildung der Risikocharakteristiken möglich.

Wichtig erscheint die Trennung von Schäden und Entschädigungen. Während Schadensumme der mönetäre Ausdruck für den tatsächlich entstandenen Schaden ist, wird unter

253 Vgl. Helten, E. (Schadensummenverteilungen, 1976), S. 113 ff.

254 Diese Risiken finden dann möglicherweise bei einem anderen Unternehmen wieder Deckung.

255 Vgl. Krengel, U. (Wahrscheinlichkeitstheorie, 1988), S. 84 f.
Diskrete Verteilungen, entstanden aus gruppierten Daten, können - Unabhängigkeit vorausgesetzt - mit relativ geringem Aufwand gefaltet werden. Damit die Zahl der Klassen nicht bei jeder Faltung steigt und deren Besetzung nicht zu sehr abnimmt, können die berechneten Verteilungen wieder ‘zurückgruppiert’ werden. Man muß dabei einen Informationsverlust hinnehmen, der jedoch für den Einsatz als Lehrinstrument keine wesentliche Rolle spielt.
Damit die Faltung von stetigen Verteilungen nicht zu aufwendig wird, werden in Object-VersPlan lediglich Exponentialverteilungen und Normalverteilungen verwendet, die bei linearen Transformationen den Verteilungstyp erhalten.

Entschädigung die vom Versicherungsunternehmen unter Berücksichtigung der Vertragsbedingungen tatsächlich bezahlte Entschädigungssumme verstanden.[256]

Für die Modellierung eines Kollektivs versicherter Bestände bedeutet dies, daß auch hier scharf unterschieden werden muß: Die Schadensimulation in jeder Periode erzeugt durch Monte-Carlo-Simulationen Schadenfälle und Schadensummen. Hieraus läßt sich das Gesamtschadenaufkommen bestimmen. Zur besseren Vergleichbarkeit kann eine Schadenquote im Verhältnis zu den Beitragseinnahmen ausgewiesen werden. Um von den Schäden zur Entschädigung zu gelangen, bedarf es der Schadenbearbeitung: Sie berücksichtigt Vertragsbedingungen und Tarifmerkrmale, was oft zu Abschlägen von der Schadensumme führen wird. Der Quotient von Entschädigungssumme und Prämieneinnahmen wird in der Praxis ebenfalls als Schadenquote bezeichnet - korrekt wäre allerdings der Begriff Entschädigungsquote.

Die Modellierung der Schadenbearbeitung wird im Abschnitt 5.3.2 (Verwaltung) noch näher beschrieben.
Die folgende Abbildung zeigt ein Beispiel für die Modellierung eines Kollektivs homogener Gewerberisiken:

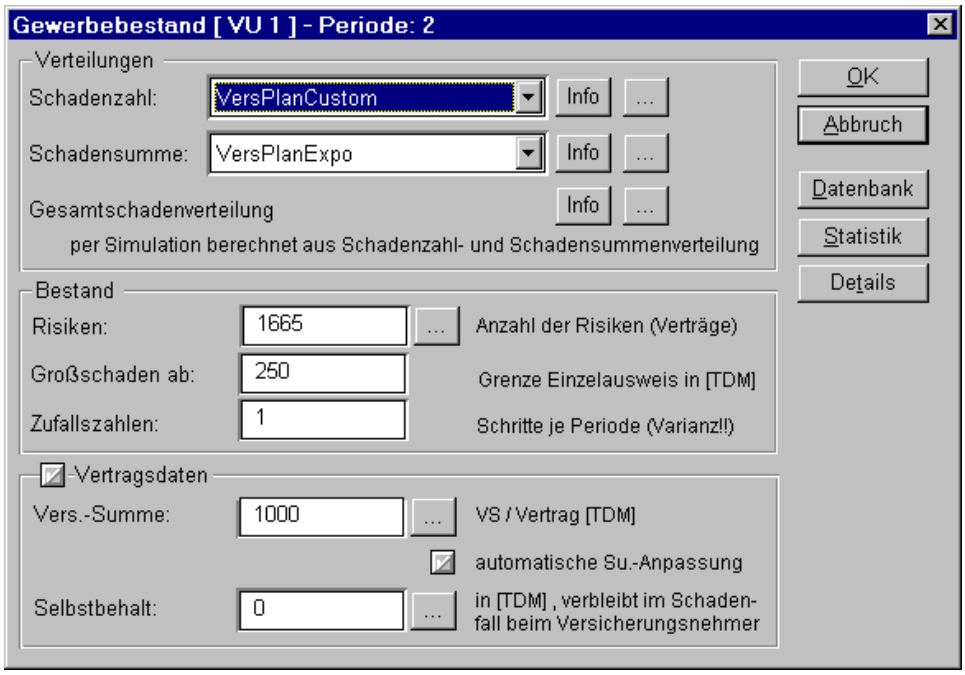



*Abbildung 33: Modellierung eines Kollektivs von Gewerberisiken*

Damit dieses Kollektiv leichter an reale Gegebenheiten angepaßt werden kann, stehen für die Beschreibung des Risikoprofils verschiedene Verteilungsfunktionen zur Verfügung:

256 Lippe, S. (Betriebskosten, 1983), S. 45

– Poisson-, Exponential- und Normalverteilung sind in der Lehre der Risikotheorie gebräuchliche Verteilungen.[257]

– Für den Einsatz eines Planspiels in der Versicherungspraxis ist die Möglichkeit von Bedeutung, beliebige diskrete Verteilungen - resultierend aus den Schadenerfahrungen - verwenden zu können.

Die Abbildung zeigt dies am Beispiel einer Schadenzahlverteilung:

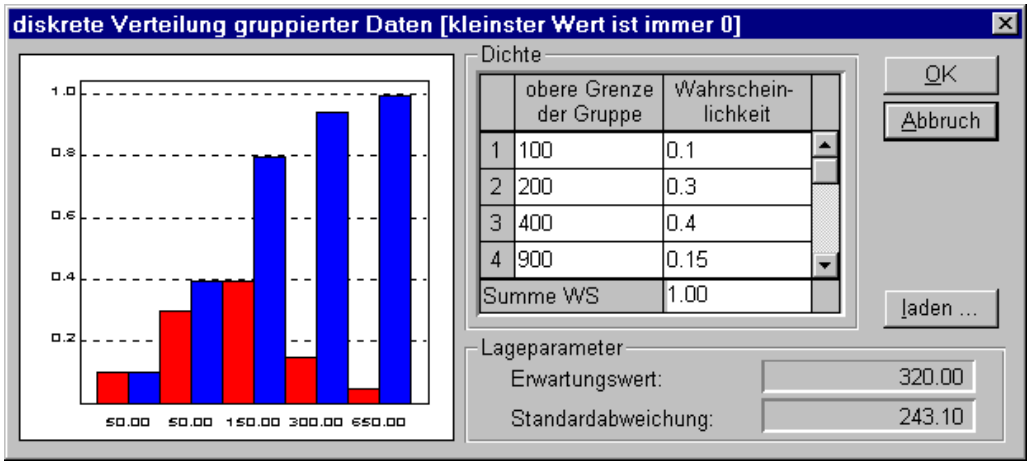



*Abbildung 34: Diskrete Schadenzahlverteilung*[258]

## Abteilungen, Mitarbeiter und Aufgaben

Eine Modellierung von Versicherungsunternehmen ist ohne die Abbildung seiner Unternehmensstruktur, der Mitarbeiter und ihrer Aufgaben nicht denkbar. Object-VersPlan bietet deshalb Klassen für Abteilungen, Mitarbeiter und deren Aufgaben.

– Abteilungen bilden den organisatorischen Rahmen für die Zusammenarbeit mehrerer Mitarbeiter.[259]

---

257 Vgl. Lippe, S. (Betriebskosten, 1983), S. 25 ff.

258 Die beiden Histogramme zeigen die Dichte und die Verteilungsfunktion. Es ist zu beachten, daß die Klassen nicht notwendigerweise die gleiche Breite besitzen müssen.

259 In der Organisationslehre werden Abteilungen als dauerhafte Gruppierung von Stellen verstanden. Stellen sind Aufgabenkomplexe, die unter normalen Umständen von einer Person bewältigt werden können. Object-VersPlan orientiert sich mit seiner Modellierung näher an der intuitiven Vorstellung und stellt den Mitarbeiter in den Mittelpunkt. Er erledigt verschiedene Aufgaben. Mitarbeiter und ihre Aufgaben bilden die Stellen, die zusammen mit der Abteilungsleitung wiederum die Abteilung bilden (siehe Abbildung ‘Unternehmen’).
Zur Organisationslehre: vgl. Picot, A. (Organisation, 1993), S. 125 ff.

Abteilungen in Object-VersPlan verfügen über eine Abteilungsleitung, die Führungsaufgaben (beispielsweise Entscheidungen zur Führung und Förderung von Mitarbeitern), jedoch keine operativen Aufgaben, wahrnehmen. Die Kapazitätsrestriktion der Führung entspricht der Beschränkung der Leitungsspanne[260].

Der Dialog zur Einstellung der Eigenschaften von Abteilungen kann in Object-VersPlan wie folgt aussehen:

*Abbildung 35: Abteilung*

– Mitarbeiter sind (ebenso wie die Risiken) als Kollektiv von Mitarbeitern modelliert: Eigenschaften und Fähigkeiten können entsprechend als Durchschnittswerte aufgefaßt werden.[261] Die Möglichkeiten zur Ausgestaltung der Mitarbeiter sind in Object-VersPlan relativ umfangreich:

  Die theoretische Grundlage für die Modellierung liefert *von Rosenstiel*: Aus den allgemeinen Bedingungen für Verhalten in Organsiationen (persönliches Wollen,

260 Vgl. Picot, A. (Organisation, 1993), S. 135 f.
Die Leitungsspannen sind stark abhängig von der Aufgabe. Deshalb bietet Object-VersPlan hierfür Möglichkeiten der Kalibrierung.

261 Es wäre als Erweiterung durchaus denkbar, individuelle Mitarbeiter (beispielsweise Außendienstmitarbeiter) zu modellieren. Zur Erhaltung der Kompatibilität mit der Makromodellierung des Mitarbeiterkollektivs müßten die Individualwerte jedoch zu Durchschnittwerten umgerechnet werden.

individuelles Können, soziales Dürfen und situative Ermöglichung)[262] definiert er Leistung als mehrdimensionale Funktion, abhängig von den Fähigkeiten und Fertigkeiten (dem individuellen Können), der Motivation (dem persönlichen Wollen und sozialen Dürfen) und der situativen Ermöglichung. *Von Rosenstiel* geht von einer multiplikativen Verknüpfung der Faktoren aus.[263] Für eine detailliertere Modellierung der personalen und organisationspsychologischen Komponente könnte man sich beispielsweise die Erweiterung um den Aspekt des Risikoverhaltens vorstellen.

Darüber hinaus werden weitere wichtige Eigenschaften für Gruppen von Mitarbeitern modelliert: Kapazität während Normalarbeitszeit und Überstunden sowie Kosten für Einstellung (Suchkosten) und Entlassung (Abfindungen). Bei Überschreitung der Arbeitszeit fallen Überlastkosten an, die man sich zum Beispiel als Kosten für zusätzliche Leih-Arbeitskräfte vorstellen könnte. Zu diesem Kapazitätsmodell können wichtige Kennzahlen (Arbeitsbelastung, Kosten, Leistungsniveau etc.) ausgewiesen werden. So werden die Parameter verändert:

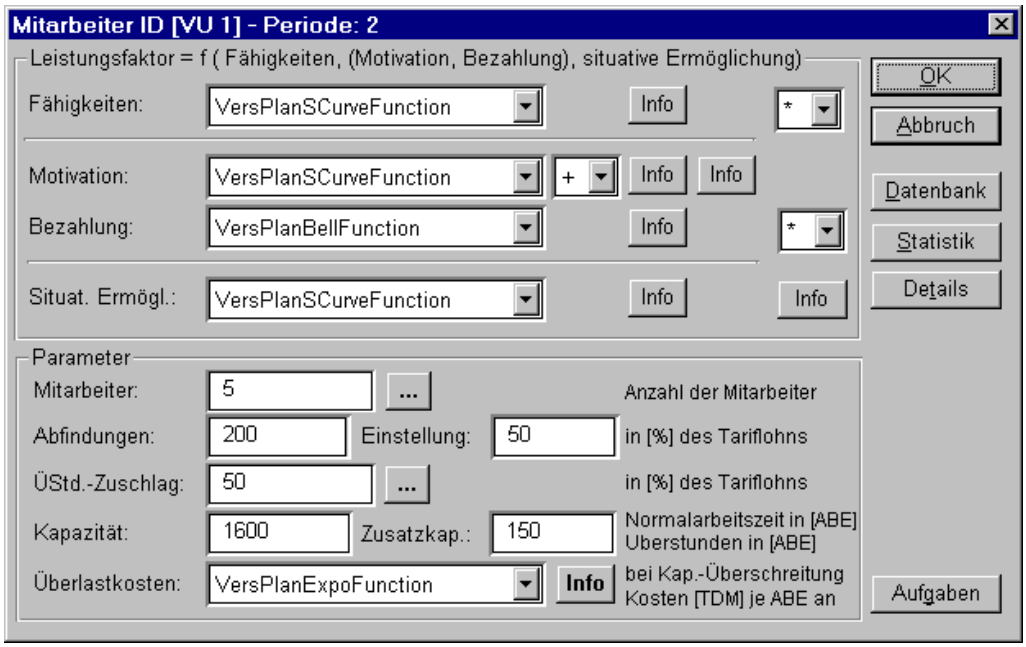



*Abbildung 36: Mitarbeiter*

Die Wirkung der Motivation und deren Einflußfaktoren wird in diesem Beispiel nur sehr grob modelliert: Es werden nur zwei Aspekte betrachtet: Allgemeine

262 Vgl. von Rosenstiel, L. (Organisationspsychologie, 1987), S. 45 f.

263 Vgl. von Rosenstiel, L. (Organisationspsychologie, 1987), S. 321ff
Nach *von Rosenstiel* entsteht Motivation aus der Interaktion von Person und Situation. Er erweitert damit die rein an psychologischen und persönlichen Aspekten orientierten Arbeiten von *Vroom* und *Lawler* um die Komponente der situativen Ermöglichung.

motivationssteigernde Faktoren und die Bezahlung.[264] Die folgende Grafik zeigt eine mögliche Modellierung der Abhängigkeit der Motivation von der Bezahlung:

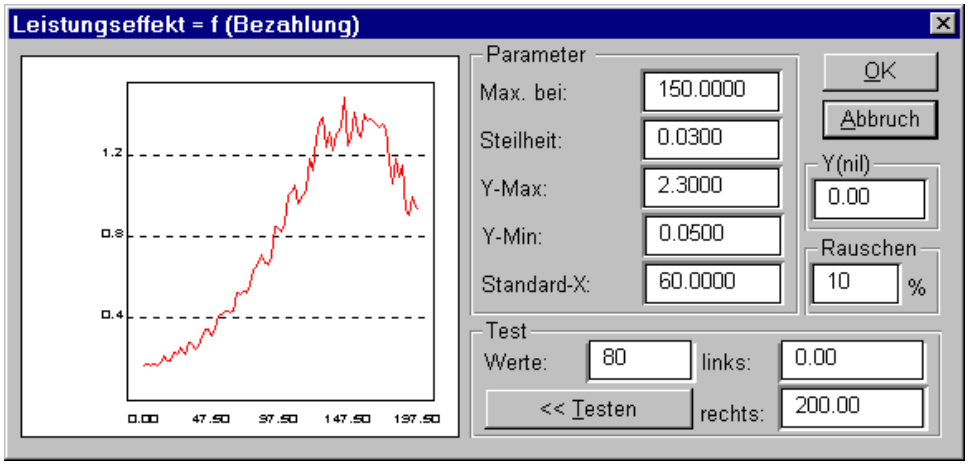



*Abbildung 37: Motivation in Abhängigkeit von der Bezahlung*

Der Leistungseffekt ist ein Beispiel für einen der vielen Anwendungsfälle von Funktionen zur Modellierung von Wirkzusammenhängen: Aus einer Palette von Funktionstypen können durch Parametervariation die passenden Funktionen generiert werden. Eine stochastische Komponente ermöglicht eine einfache Abbildung von Unschärfen in Gestalt von Rauschen. [265]

Für Erweiterungen ist gerade hier ein weiterer Spielraum gegeben. In diesem Zusammenhang sei auf die Arbeiten zur Motivationspsychologie und auf die Theorie der Leistungsmotivation verwiesen.[266]

– Die Mitarbeiter führen Aufgaben wie zum Beispiel Schadenbearbeitung oder Antragsbearbeitung aus. Je nach Umfang der Tätigkeit fällt die Modellierung detailliert oder einfach aus. Allen Aufgaben gemein ist die Möglichkeit der Berücksichtigung der Qualität der Aufgabenerledigung durch die Mitarbeiter (die sich auf die Effektivität niederschlägt) und die optionale Berücksichtigung von Transaktionskosten.[267]

---

264 Zur Wirkung von Entlohnung auf Motivation und Leistung: vgl. von Rosenstiel, L. (Organisationspsychologie, 1987), S. 345 ff.

265 Durch das wiederverwendungsorientierte Konzept von Object-VersPlan steht für die Modellierung der Abhängigkeiten die ganze Palette von Funktionen zur Verfügung: Glockenfunktionen, Sigmoid-Funktionen, Exponentialfunktionen, Polynom-Funktion und Kombinationen daraus. Gleichzeitig stehen pro Funktion bis zu fünf Parameter zur individuellen Anpassung der Lage und der Form zur Verfügung. Erweiterungen dieser Funktionalität werden an allen Stellen ohne weitere Änderungen verfügbar.

266 Vgl. von Rosenstiel, L. (Organisationspsychologie, 1987), S. 183 f.; vgl. a. ders. S. 188 ff.

267 Zum Begriff Transaktionskosten: vgl. Picot, A. (Organisation, 1993) S. 106 f.

Durch die Trennung von Mitarbeitern und Aufgaben in zwei Objektklassen ist eine flexible Gestaltung von Organisationen möglich. Das Beispiel Sachbearbeitung im Innendienst von Versicherungen mag dies verdeutlichen: In einem Fall wird in klassischer Weise die Schadenbearbeitung und die Bearbeitung von Neuanträgen getrennt vorgenommen, im anderen Fall führt ein Gruppe von Universalsachbearbeitern beide Aufgaben durch. Zur Modellierung werden jeweils nur die Objektklassen Abteilung, Mitarbeiter und Aufgabe benötigt - es ist nicht notwendig, Schadenbearbeiter und Universalsachbearbeiter unterschiedlich zu implementieren.

Die folgende Abbildung zeigt die Aufgabe ‘Schadenbearbeitung’:

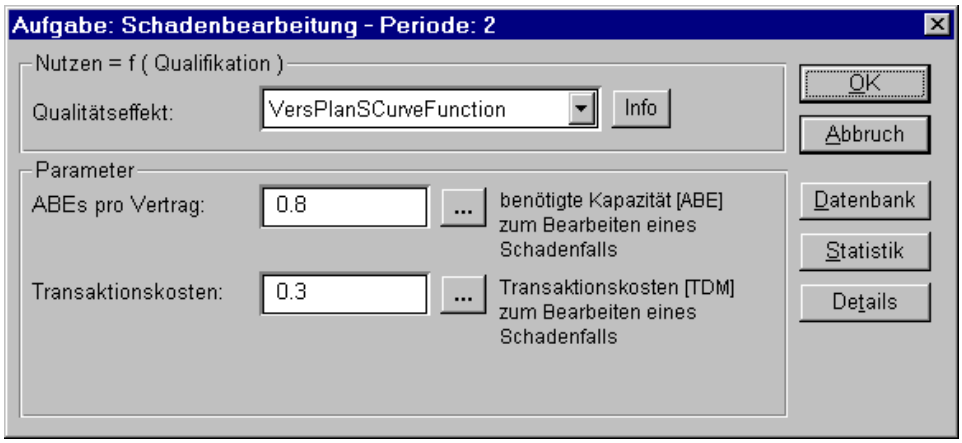



*Abbildung 38: Schadenbearbeitung*

Interessant ist die Betrachtung des Qualitätseffekts: Damit soll beispielsweise ausgedrückt werden, wie verbesserte Qualifikation des ausführenden Mitarbeiters die Qualität der Arbeit beeinflußt. Die Art der Beeinflussung sieht natürlich bei jeder Tätigkeit ein wenig anders aus: Im Fall der Schadenbearbeitung ist Qualität als Exaktheit der Schadenprüfung modelliert.[268]

## Spielleiter

Wie bereits eingangs angedeutet, erfüllt der Spielleiter die Funktion der Externalität. Er gibt die Rahmenbedingungen für die Simulation vor: In dem vorliegenden Beispiel ist dies die Startzahl des Zufallsgenerators[269] und die Kennzahl für die gesamtwirtschaftliche Lage. Hier wären zahlreiche Erweiterungen denkbar wie zum Beispiel Inflationsrate,

268 Hier zeigt sich deutlich die Überlegenheit der Trennung von Mitarbeiter und Aufgaben: Mit jeder neu betrachteten Dimension (in diesem Beispiel also der Qualitätseffekt) würde sich bei gemeinsamer Modellierung die Zahl der notwendigen Varianten multiplikativ erhöhen.

269 Da die Erzeugung von Zufallszahlen algorithmisch geschieht, ist die Startzahl des Zufallsgenerators wichtig für die Reproduktion von Ergebnissen: Gleiche Startzahlen führen bei identischen Ausgangsbedingungen zu den gleichen Resultaten.

Wirtschaftswachstum[270], besondere Ereignisse wie Naturkatastrophen und ähnliches, das die Seminarsituation entsprechend der Erfordernisse beeinflussen kann.

Es steht dem Spielleiter prinzipiell auch die Möglichkeit offen, die Verhaltensweisen (ausgedrückt durch Reaktionsfunktionen) der an der Simulation beteiligten Objekte zu verändern. In der Praxis ist jedoch nur eine sehr behutsame Anwendung dieser Eingriffsmöglichkeit angezeigt. Sie nimmt den Teilnehmern die Möglichkeit, Modellzusammenhänge zu erkennen und die Anwendung und Korrektur eigener Strategien zu erproben. Leichte Änderungen können gerechtfertigt sein - entsprechen sie doch beispielsweise den sich langsam verändernden Verhaltensweisen und Werten in der Gesellschaft.[271]

Der Spielleiter hat noch eine weitere Funktion: Er kontrolliert den Simulationsablauf. In den nun folgenden Abschnitten über die Modelldynamik wird diese Aufgabe klar.

## 5.3.2. Ablauforganisation

Alle im vorangegangenen Abschnitt vorgestellten Systemelemene wirken während der Simulation einer Spielperiode in einem Geflecht hierachisch angeordneter Abläufe zusammen: Eine Ablauforganisation bildet die qualitative Dynamik des Modells ab.

### Ablauf einer Simulationsperiode

Um die Kompliziertheit der Modellierung nicht unnötig zu erhöhen, wurde von der Möglichkeit der Zerlegung der Modellierung der Ablauforganisation in Submodellen ausgiebig Gebrauch gemacht. Oberste Ebene ist der Ablauf einer Simulationsperiode[272]:

270 Zum Einfluß von Wirtschaftswachstum und Konjunktur auf das Versicherungsgeschäft: vgl. Helten, E. (Konjunktur, 1983), S. 1190 ff.

271 Zum Wertewandel und die Einflüsse auf die Versicherungswirtschaft siehe Harbrücker, U. (Wertewandel, 1992)

272 In diesem Beispiel entspricht die Simulationsperiode einem Geschäftsjahr. Es ist aber durchaus denkbar, Quartale, Monate oder gar Wochen und Tage als Periodenlänge zu wählen. Dabei ist zu beachten, daß sich alle Mengenangaben in allen an der Simulation beteiligten Objekten auf diese Periodenläge beziehen. Soll also die Periodenlänge verändert werden, müssen auch alle Einstellungen angepaßt werden.

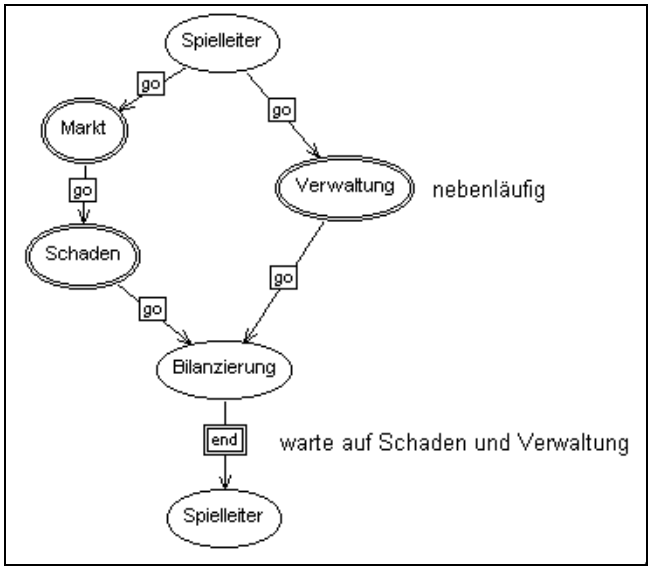



*Abbildung 39: Ablauf einer Periode*

Wie die Abbildung zeigt, sind in Object-VersPlan nebenläufige Aktionen denkbar. In obigem Beispiel werden die Verwaltungstätigkeiten parallel zur Markt- und Schadensimulation ausgeführt:

1. Am Anfang steht die Bestandsbetreuung durch die Außendienstmitarbeiter der Versicherungsunternehmen.

2. In der Marktsimulation werden einerseits neue Bestände akquiriert und andererseits gehen Bestände durch Storno zurück in den Markt. Dementsprechend sind jetzt die Tätigkeiten der Antragsbearbeitung zu erledigen.

3. Erst dann kann die Schadensimulation laufen.

4. Sind die Schäden simuliert, kann die Schadenberbeitung durchgeführt werden.

5. Wenn sowohl die Markt- und Schadensimulation als auch die Verwaltungsaufgaben erledigt sind, können Bilanzierungsarbeiten zur Erstellung des Periodenabschlusses durchgeführt werden.

Alle Aktionen finden in der selben Simulationsperiode statt. Quantitative Dynamik entsteht erst durch die Durchführung mehrerer Spielperioden: Änderungen der Eigenschaften und des Verhaltens der Systemelemente und der Struktur der Aufbauorganisation können im Vergleich der Perioden beobachtet werden.

Die nun folgenden Abschnitte zeigen Beispiele für die Implementierung dieser Teilaufgaben in aufsteigender Komplexität - und nicht in der Reihenfolge der Abarbeitung.

### Schadensimulation

Die Schadensimulation ist ein sehr einfaches Beispiel für Abläufe: Das Submodell zum Objekt Schadensimulation zeigt, daß lediglich in allen Beständen (vgl. 5.3.1) die Methode zur Schadensimulation angestoßen wird:

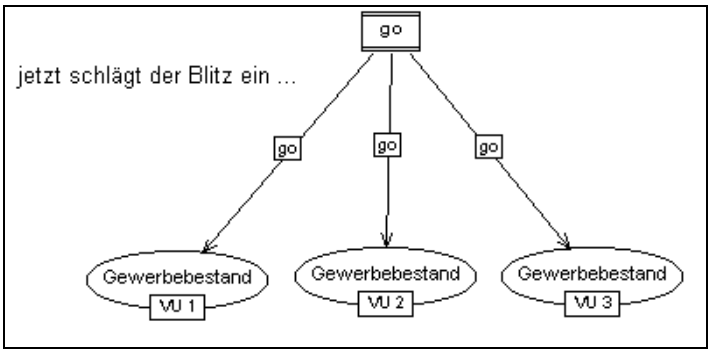



*Abbildung 40: Schadensimulation*

Die Objekte der Gewerbebestände führen nach Erhalt der Nachricht ‘go’ die Schadensimulation durch: Den Verteilungen von Schadenzahl und Schadenhöhe folgend werden per Monte Carlo-Simulation Schäden simuliert und protokolliert. Nach der Ausführung dieser drei Aktionen wird der Simulationsablauf eine Ebene höher mit der Bilanzierung fortgesetzt.

### Marktsimulation

Eine Stufe komplexer ist die Marktsimulation.Gerade an diesem Anwendungsbeispiel zeigt sich die enorme Mächtigkeit und Flexibilität des objektorientierten Ansatzes. Doch zuerst zur Idee:

- In einem Versicherungsplanspiel wird das immaterielle Gut Versicherungsschutz produziert und am Markt abgesetzt. Es wird ein Wettbewerb über Preise (dem Tarifniveau, ausgedrückt in Prozent von der Versicherungssumme oder in absoluten Beträgen) und Konditionen (Tarifbedingungen etc.) geführt. In sofern unterscheidet sich ein Versicherungsmarkt nicht von einem Markt für materielle Güter.

- Märkte für Versicherungsprodukte zeichnen sich unter anderem jedoch durch einige Besonderheiten aus, die auch bei der Modellierung in einem Versicherungsplanspiel berücksichtigt werden müssen: Versicherungsschutz ist an einen Versicherungsgegenstand gebunden, der die Eigenschaft der Versicherbarkeit erfüllen muß - im folgenden mit Risiko bezeichnet. Innerhalb eines Marktes gibt es, je nach wirtschaftlicher Situation, eine bestimmte Anzahl von Risiken, die im Modell als Marktpotential bezeichnet werden. Die Eigner dieser Risiken stehen nun vor der

Entscheidung, ob sie dieses Risiko selbst tragen[273] oder ob sie es einem Versicherungsunternehmen in Deckung geben werden.[274] Wird ein Risiko in Deckung genommen, so hat dies einen Vertrag zur Folge. Im Modell werden nur einjährige Verträge mit automatischer Verlängerung betrachtet. Der Wettbewerb führt dazu, daß Verträge gekündigt, daß Risiken bei anderen Unternehmen zu besseren Bedingungen plaziert oder daß sie auch gar nicht mehr versichert werden: Im Modell bedeutet dies, daß Teile bisher versicherter Risiken (hier Bestände genannt) in das Marktpotential zurückgehen, dort unversichert verbleiben oder in den Bestand eines anderen Versicherungsunternehmens aufgenommen werden. Diese Risiken werden - aus Sicht des abgebenden Versicherungsunternehmens - mit Storno bezeichnet.

- Zu diesem Umschichtungsprozeß von Risiken kommt noch eine versicherungsspezifische Komponente hinzu: Der Qualitätsaspekt von Risiken. Ob ein Preis für die Versicherung eines Risikos profitabel ist oder nicht, hängt maßgeblich von dessen potentiellem Entschädigungsaufkommen ab. Hierunter subsummieren sich im wesentlichen die Entschädigungszahlungen für Schadenfälle, aber auch zu Unrecht in Anspruch genommene Entschädigungen (beispielsweise durch Versicherungsbetrug). Jeder Versicherer ist nun interessiert - gemessen an seinen Prämieneinnahmen - wenig entschädigungsträchtiges Geschäft zu akquirieren. Deshalb widmet er sich der Risikoselektion, um die ‘guten Risiken’ aus dem Marktpotential in seinen Bestand zu holen.

- Damit die Vorgänge der Marktsimulation für die Lernenden nachvollziehbar bleiben, werden im Modell homogene Risiken zu gleichen Bedingungen versichert. So wird beispielsweise auch nur eine durchschnittliche Versicherungssumme ausgewiesen.

- Für das hier vorliegendene Modell wird davon ausgegangen, das keine prinzipiellen Restriktionen bei der Zeichnungskapazität (entspricht der Kapazitätsbeschränkung auf Märkten knapper Güter) bestehen: Alle versicherbaren Risiken, die von den Nachfragern zu bestimmten Konditionen (Tarifmerkmale, Preis) versichert werden sollen, können bei den Versicherungsunternehmen untergebracht werden.

- Es werden Teilmärkte betrachtet: Für jede Region und jedes Produkt kann eine individuelle Marktsimulation durchgeführt werden: Durch die Berücksichtigung spezifischer Gegebenheiten wie Marktpotential, Ausschöpfung und Wachstumsraten sowie unterschiedliche Eigenschaften und Verhalten der Nachfrager können völlig verschiedene Märkte betrachtet werden. In der vorliegenden Version beeinflussen sich die Märkte noch nicht gegenseitig.

---

273 Wenn hier die Rede davon ist, daß der Versicherungsnehmer das Risiko selbst trägt, so sind damit alle Möglichkeiten der Absicherung mit Ausnahme von Versicherungsschutz gemeint.

274 Dies entspricht etwa der mikroökonomischen Betrachtung der Präferenzordnung zwischen zwei substitutiven Gütern.

Die folgende Abbildung zeigt den prinzipiellen Ablauf der Marktsimulation:

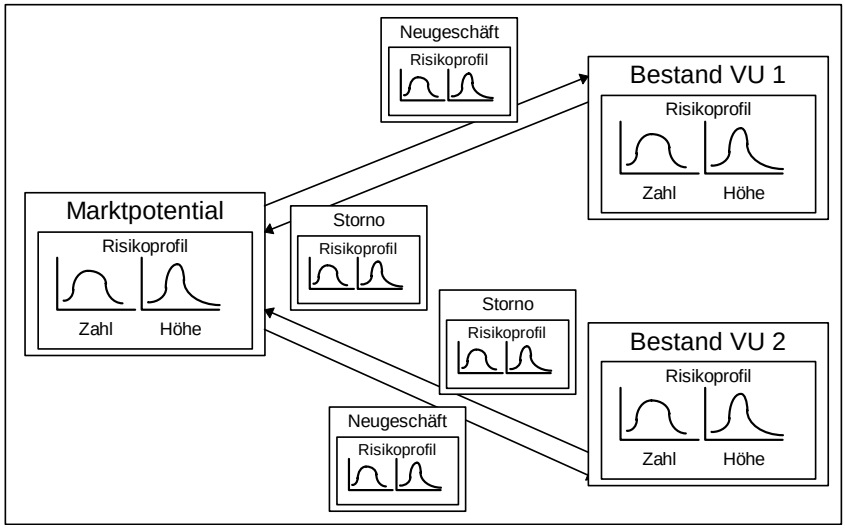



*Abbildung 41: Ablauf der Marktsimulation*

Damit diese Umschichtungsprozesse verwaltet werden können, bedient sich Object-VersPlan des Mechanismus des Auktionators (vgl. 5.3.1):

- Der Auktionator versieht die Aufgabe der Vergabe von Marktanteilen wie eine unsichtbare Hand: Er weiß über die Präferenzen der Marktteilnehmer (Anbieter und Nachfrager) Bescheid und holt sich in jeder Periode aktuelle Informationen, wie beispielsweise Preise, Aufwendungen für Werbung und Außendienst.

- Aufbauend auf diesen Informationen ermittelt er Neugeschäft und Storno.

- Für jeden Teilmarkt wird ein eigener Auktionator aktiv: Damit können Märkte mit unterschiedlichen Eigenschaften modelliert werden.

Die nächste Abbildung zeigt die Dialogbox zur Einstellung der Präferenzen und Verhaltensfunktionen:

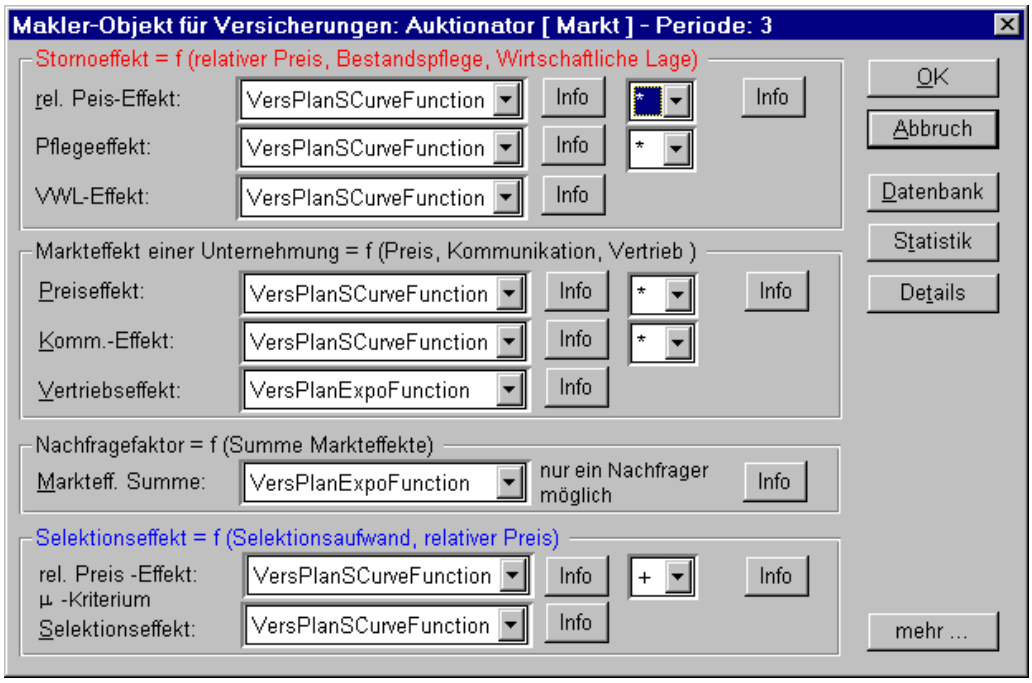



*Abbildung 42: Auktionator*

Zentrum der Modellierung ist die Abbildung des Markteffektes eines Anbieters durch eine mehrdimensionale Funktion in Abhängigkeit vom Angebotspreis, den Kommunikationsaufwendungen (Werbung etc.) und den Anstrengungen des Aussendienstes. Standardmäßig ist eine multiplikative Verknüpfung vorgesehen.[275]

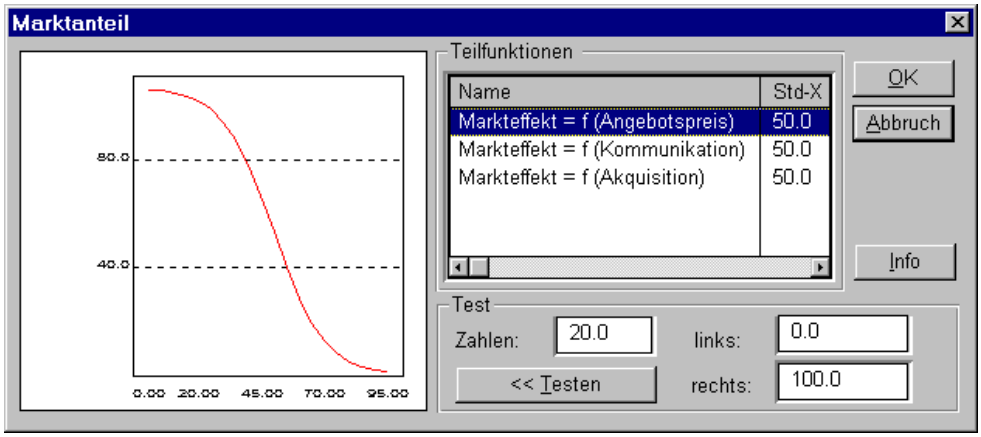



*Abbildung 43: Modellierung der Markteffekte*

275 Zur Modellierung von Marktreaktionsfunktionen: vgl. Hannsmann, F. (Quantitative Betriebswirtschaftslehre, 1985), S. 113 ff.

Die Grafik zeigt einen Längsschnitt durch die mehrdimensionale Marktanteilsfunktion zur Darstellung der Abhängigkeit von Markteffekt vom Angebotspreis.

Es müssen nicht alle im obigen Beispiel angebotenen Funktionen benutzt werden. Für einfache, weniger komplexe Modelle kann der Auktionator auch nur mit der Preis/Absatz-Funktion benutzt werden. Der Auktionator erkennt, wenn beispielsweise keine Werbung oder kein Außendienst vorhanden sind und benutzt dann die entsprechenden Funktionen nicht. Durch die Modellierung relativer Markteffekte ist es leicht möglich, zusätzliche Anbieter einzufügen, ohne die Funktionen anpassen zu müssen. Auktionatoren führen eine Statisktik über die Marktvorgänge. Die Abbildung zeigt einen Auschnitt der verfügbaren Kennzahlen:

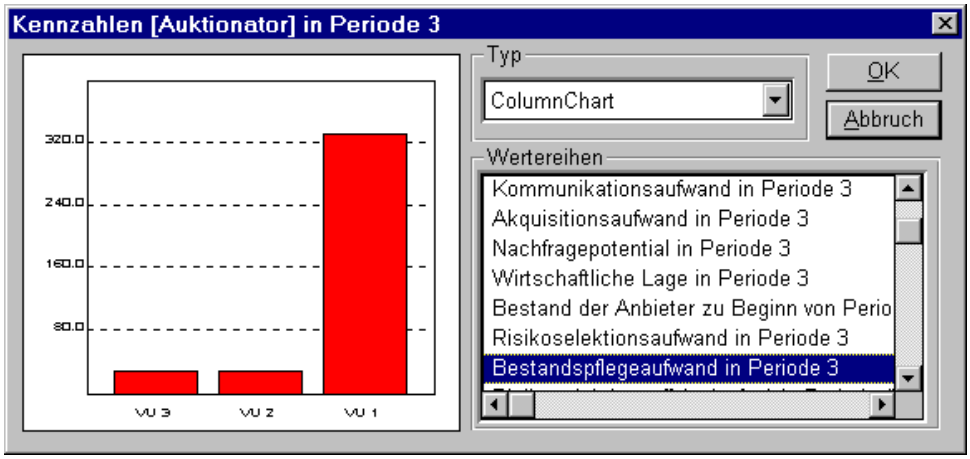



*Abbildung 44: Kennzahlen des Auktionators*

Diese Kennzahlen können in Berichten verwendet werden: Marktberichte geben den Teilnehmern einen Überblick über das Marktgeschehen. Spiele-Konstrukteure erhalten eine Möglichkeit zur Abstimmung der Verhaltensfunktionen.

Zur besseren Übersicht wurden die Simulationsabläufe zum Markt auf mehrere Ablaufmodelle verteilt: Angebot, Nachfrage, Storno und Bestandspflege. Die folgende Abbildung zeigt eines dieses Teilmodelle:

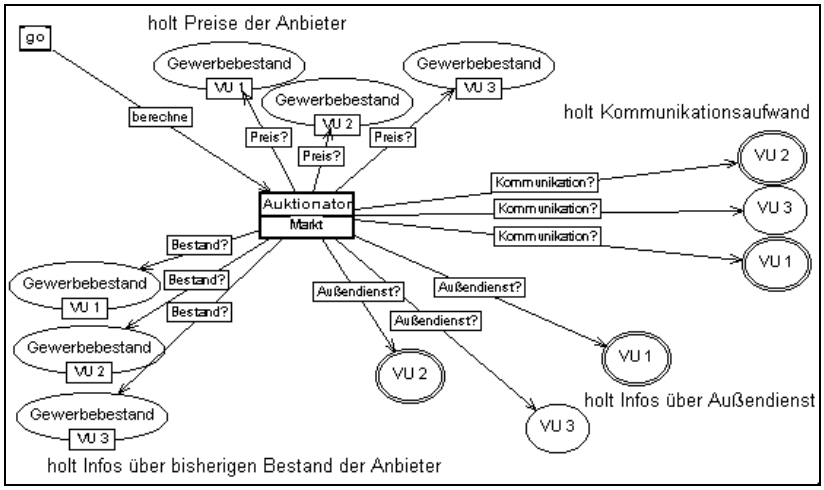



*Abbildung 45: Marktsimulation: Angebot*

## Verwaltung

Die Simulationsabläufe zur Verwaltung machen von den Möglichkeiten bedingter, nebenläufiger Prozesse Gebrauch.

Verwaltungsaufgaben werden getrennt für jedes Unternehmen simuliert: Dementsprechend ist jedem Unternehmen auch ein eigenes Ablaufmodell zugeordnet. Diese Modelle werden als Submodelle des Objekts ‘Verwaltung’ in parallelen Prozessen neben der Schadensimulation und der Marktsimulation abgearbeitet.

Die folgende Abbildung zeigt den Simulationsablauf für die Tätigkeiten des Innendienstes (=Verwaltung) des Unternehmens VU 1:

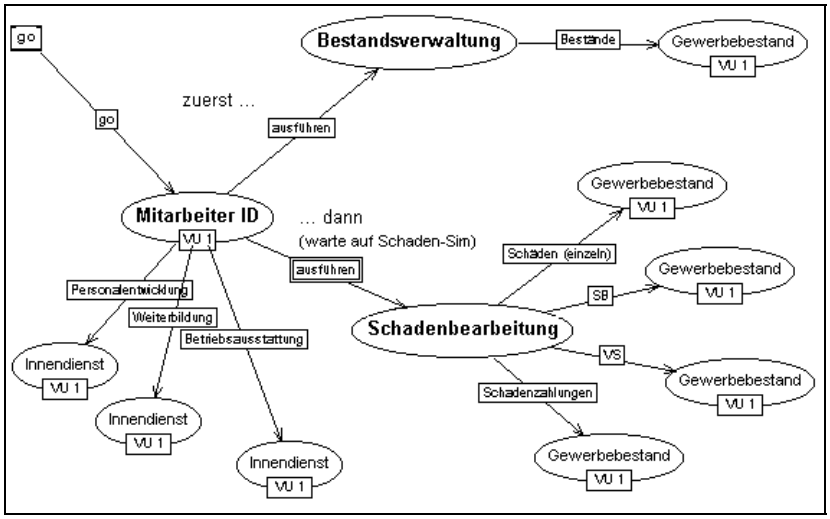



*Abbildung 46: Verwaltung (VU 1)*

Das Beispiel zeigt zum einen das Zusammenspiel der Objekte Mitarbeiter und Aufgaben und benutzt zum anderen die Möglichkeiten, Reihenfolgen und Bedingungen für die Ausführung von Aktionen zu festzulegen:

- Die Aufgabe Bestandsverwaltung wird vor der Schadenbearbeitung ausgeführt.

- Die Schadenbearbeitung wird erst dann ausgeführt, wenn die Schadensimulation abgeschlossen ist - also überhaupt Schäden zum Bearbeiten vorhanden sind.

Wenn man Reihenfolgen und Bedingungen festlegt, ist die Gefahr gegeben, daß die Simulation in einen Zustand gerät, der eine Weiterführung verhindert: Zwei Ereignisse bedingen einander gegenseitig - aber keines der beiden wurde bisher ausgeführt. Object-VersPlan bietet bei der Erstellung der Verknüpfungen Plausibilitätsprüfungen, die solche dead-locks verhindern helfen sollen. Werden sie nicht schon bei der Konstruktion entdeckt, kann die Simulation unterbrochen werden. Das Ereignisprotokoll und der Debugger helfen dann, den Fehler zu entdecken.

### Dynamik durch veränderte Aufbauorganisation

Damit die Komplexität für anspruchsvollere Anwendungsfälle noch weiter gesteigert werden kann, muß die Dynamik der Modellierung um einen zusätzlichen Aspekt erweitert werden: Aufbauorganisationen sind immer nur Zeitpunktbetrachtungen. Ändern sich die Rahmenbedingungen und die Anforderungen, so muß sich auch die Aufbauorganisation anpassen können. Im Falle eines Unternehmensplanspiels kann dies durch die Spielleitung oder auch durch Entscheidungen der Spieler geschehen. Anhand von zwei kurzen Beispielen soll ein Ausblick zeigen, wie dies in Object-VersPlan berücksichtigt werden könnte. Beides sind Beispiele von „langsamer Makrodynamik“[276], die mit komparativ-statischer Betrachtungsweise erfaßt werden können:

- Produktinnovation: Unternehmen bringen ein zusätzliches Produkt auf den Markt. Um Bestandteil der Simulation zu werden, muß das neue Produkt in der Teilehierarchie des Unternehmens verankert werden. Damit es auf einem Markt gehandelt werden kann, müssen ein Auktionator und entsprechende Ablauforganisationen für dieses Produkt erzeugt werden.

- Organisationentscheidungen: Sehr viel komplexer ist die Einbeziehung dynamischer Aufbauorganisationen von Unternehmen. Die Spieler können über Organisationsformen ihres Unternehmens entscheiden (Sparten- oder Kundengruppenorientierung? Universal- oder Spezialsachbearbeitung?). Bisher wurden nur Entscheidungen zugelassen, die sich auf Eigenschaften oder Verhaltensannahmen beziehen - jetzt soll eine Entscheidung auf die Struktur wirken.

276 Rapoport, A. (Systemtheorie, 1988), S. 67

Damit Umorganisationen in der bisherigen Modellstruktur von Object-VersPlan möglich sind, müssen Teilebeziehungen verschoben werden können. Dies ist ein leicht zu lösendes Problem. Kommen neuen Objekte hinzu (Bsp.: Eine Abteilung wird wegen Überlastung geteilt), so müssen diese Objekte in (neue) Ablaufmodelle einbezogen werden - ein noch zu lösendes Problem. Gelöschte Objekte werden ohnehin automatisch auch aus den Abläufen entfernt.

Diese Änderungen der Modellstruktur sind in Object-VersPlan bisher nocht nicht automatisch möglich - sie müssen vom Spielleiter manuell durchgeführt werden.

# 6. Zusammenfassende Bewertung und Ausblick

Dieses abschließende Kapitel versucht, die während der vorangegangenen fünf Kapitel diskutierten Konzepte und Ideen zusammenfassend zu bewerten und wagt einen Blick in die Zukunft von Planspielen und Planspielentwicklung.

- Hierzu wird im ersten Abschnitt anhand der erarbeiteten Kriterien überprüft, inwieweit sich der objektorientierte Ansatz zur Planspielentwicklung eignet.

- Die Konzepte objektorientierter Planspielentwicklung weisen Bezüge zu anderen Anwendungsfällen auf. Der zweite Abschnitt stellt Ansatzpunkte und Implikationen für die Nutzung von Synergien her.

- Im letzten Abschnitt schließlich sollen aus Sicht der Beschäftigung mit Planspielen und Planspielentwicklung Herausforderungen für die Zukunft aufgezeigt werden.

## 6.1. Eignung des objektorientierten Ansatzes zur Planspielentwicklung

Zum Abschluß der Betrachtungen ist selbstverständlich zu fragen, ob der objektorientierte Ansatz den Erwartungen an die Planspielentwicklung gerecht geworden ist. Hierzu sei noch einmal zusammenfassend auf die im dritten Kapitel erarbeiteten Kriterien zurückgegriffen:

### Objektorientierte Modellbildung

Der objektorientierte Ansatz ist mit dem Anspruch angetreten, die semantische Lücke zwischen Realitätsausschnitt und Modell zu schließen. Die Prinzipien der Objektorientierung sind mit nur wenigen Punkten beschrieben und intuitiv einsichtig. Es entsteht ein Modell, das der natürlichen, systemorientierten Denkweise sehr nahe kommt. Die der Objektorientierung inhärenten Möglichkeiten zu Klassifikation und Bildung von (vernetzten) Teilmodellen können trotz aller Komplexität die Kompliziertheit in Grenzen halten. Dabei liegt die große Stärke darin, daß alle Beteiligten eine gemeinsame Sprache sprechen: Der Entwicklungsprozeß kann sehr viel interaktiver und mit qualitätsfördernden Rückkoppelungen zwischen allen Beteiligten betrieben werden.

Mit der Anwendung der Objektorientierung allein sind die prinzipiellen Probleme der Modellbildung noch nicht gelöst - es wird dem Modellierenden allerdings nicht noch zusätzlich schwerer gemacht. Aus Sicht des (nicht programmierenden) Anwenders und

Veranstalters, der ein anforderungsgerechtes Planspielmodell konstruieren möchte, helfen wiederverwendbare Komponenten, das Modellierungsproblem in den Griff zu bekommen. Eine anwendungsorientierte Klassenbibliothek legt durch ihr Design die Sichtweise des Realitätsausschnitts de facto fest: Eine optimale Funktionsweise im Sinne der wiederverwendbaren Komponenten bedingt, daß man bei der Verwendung dem Gesamtkonzept - den aufeinander abgestimmten Vernetzungsmöglichkeiten und Verhaltensannahmen - des Entwicklers folgt; diese gilt es zuerst einmal zu erfassen. Resignierend könnte man meinen, die Erkenntnisproblematik der Modellbildung sei nur um eine - die Realität eventuell verfälschende - Stufe verschoben worden. Doch wie bereits ausgeführt wurde: Planspielentwicklung ist ein interaktiver Prozeß - Entwickler und Anwender werden das Modell gemeinsam erarbeiten und eine gemeinsame, den Anforderungen der Aufgabe entsprechende Sicht des Realitätsauschnittes für das Planspiel-Simulationsmodell konzipieren. Unterstützt wird dies auch durch Organisationsformen der Zusammenarbeit wie strategische Netzwerke und Kooperationen.

Im Mittelpunkt des Planspiels steht der Anwendungskontext - derjenige Ausschnitt aus der Realität, in den die Lernsituation eingebettet ist: Die Gestaltung dieses Kontextes ist vor allem in Hinsicht auf die Komplexität der Modellierung einer der zentralen Aspekte anforderungsorientierter Planspielentwicklung. Die Beschäftigung mit diesem Thema hat gezeigt, daß es letztlich die modelltheoretischen Beschränkungen im Spannungsfeld zwischen Mikro- und Makromodellierung sind, die eine beliebige Variation der Komplexität verhindern. Für die Planspiel-Praxis hingegen scheint diese Diskussion nicht in gleichem Maße relevant: Die beispielweise in Object-VersPlan verfügbaren Möglichkeiten zur Anpassung des Modells sind bereits so vielfältig, daß sie weit besser als bisher verfügbare Planspielimplementierungen lernziel- und zielgruppenspezifische Modelle erlauben. Aus dem Blickwinkel der Effizienz (der Lerneffekte) eines Planspiels erscheint es zweifelhaft, ob besonders detaillierte Modelle, wie beispielsweise Mikroverhaltensmodelle, überhaupt dazu geeignet sind, die so wichtigen Fähigkeiten der Systemkompetenz zu vermitteln.

Der dritte wesentliche Aspekt behandelt die Möglichkeiten zur Modelldokumentation und zur Interaktion mit dem Modell. Der objektorientierte Ansatz bietet mit M-V-C ein überaus mächtiges Konzept, das besonders in Hinsicht auf die Anwenderorientierung und die Mobilisierung der Wiederverwendungspotentiale seine Stärken hat.

### Unterstützung der Lernkonzepte

Ein Ansatz zur Planspielentwicklung ist unbrauchbar, wenn er nicht die für Planspiele zentralen Lernkonzepte unterstützt. Der hier vorgestellte Ansatz der objektorientierten Planspielentwicklung ist gerade vor dem Hintergrund dieser Forderung entstanden. Es sollte gezeigt werden, daß es die Objektorientierung ermöglicht, Planspiele für verschiedene Lernsituationen zu konstruieren.

Aufbauend auf den praktischen Erkenntnissen mit klassischen Planspielseminaren wurde eine Erweiterung sowohl der Anwendungssituationen durch Erhöhung der Interaktivität mit dem Simulationsmodell als auch der Einbeziehung dynamischer Modellkomponenten konzipiert. Verschiedene Probleme haben dabei die praktische Realisierung behindert:

Modelle, die die klassische Seminarsituation unterstützen, sind in der Regel Makro-Modelle in komparativ statischer Betrachtung; sie lassen sich durch Verkürzung der Periodenlängen in quasi-stetige Modelle überführen. Damit ihre Aussagekraft jedoch nicht verloren geht, darf die Periodenlänge nicht beliebig kurz gewählt werden. Für den Einsatz in Unternehmensplanspielen ist es keine gravierende Einschränkung, wenn beispielsweise nur Monatsscheiben simuliert werden. Eine ‘stufenlose’ Überführung von Makro-Modellen in Mikroverhaltensmodelle ist aus Kompatibilitätsgründen (Aggregationsproblematik, fehlende Mikro-Fundierung) und wegen der unterschiedlichen Anforderungen an die Simulationsumgebung nicht realisierbar. Dynamische Änderungen der Modellstruktur sind noch nicht möglich, da sich ein entsprechendes Konzept noch in Entwicklung befindet.

Die zum heutigen Zeitpunkt mangelnde technische Reife und die relativ schlechte Leistung objektorientierter Datenbanken bei Transaktionen, die aus sehr vielen, komplexen und geschachtelten Datenbankzugriffen bestehen, macht die theoretisch verfügbare Unterstützung quantitativ dynamischer Systeme in der Praxis noch nicht verwendbar.[277]

Insgesamt betrachtet aber kann Objektorientierung die pädagogische Fundierung von Planspielen erheblich unterstützen und erweitern. Sie bietet praktikable Lösungen für die Forderungen konstruktivistisch fundierter Lernansätze:

- Marktsimulationen unterstützen den Wettbewerbscharakter zur Mobilisierung intrinsischer Motivation. Durch flexible Möglichkeiten zur Variation des Marktverhaltens (der Marktreaktionsfunktionen) sowie der Komplexität der Märkte und Teilmärkte steuert der Spielleiter die Wettbewerbsintensität und die Eingriffsmöglichkeiten der Teilnehmer in weitem Umfang.

- Authentische Lernsituation können durch die Kombination von realitätsnah konstruiertem Simulationsmodell und der Handlungssituation als Spiel-Unternehmer verwirklicht werden.[278] Die oben zusammengefaßten Stärken der Modellierung und die Verfügbarkeit von Objektbibliotheken helfen, dies zu erreichen. Dynamik erhält die Planspielsituation nicht allein durch das Simulationsmodell - es sind

277 Diese Einschränkung gilt in solcher Schärfe nur für PC-basierte Systeme wie das bei Object-VersPlan zum Einsatz gekommene Datenbanksystem VC ODBMS. Mit zunehmender Leistung der PCs und durch weiterentwickelte Optimierungsmethoden der Datenbanksysteme wird sich dieses Problem verringern.

278 So bemerkte einmal ein Teilnehmer des Versicherungsplanspiels zu seiner Rolle als Vorstand seines Spiel-Unternehmens: ‘Das ist gar kein Spiel. Das ist ja ernst. Hier geht es viel emotionaler und umkämpfter zu, als in der Realität an meinem Arbeitsplatz!’

gleichermaßen die dynamikerzeugenden Prozesse, die bei Verhandlungen zwischen den Teilnehmern in und zwischen den Gruppen ablaufen.

– Modelle, die breit anstatt tief angelegt sind, bieten die Möglichkeit zur Vermittlung multipler Perspektiven und unterschiedlicher Kontexte.

– Objektorientierte Werkzeuge, die (dem M-V-C-Ansatz folgend) modular an den Bedürfnissen der Benutzer orientiert sind, bieten interaktive Dokumentationsmöglichkeiten. Sie unterstützen den Lernenden bei der Artikulation von Gelerntem, und bieten Hilfestellungen, eigene Entscheidungen selbst zu überwachen und zu kontrollieren.

## 6.2. Bezüge objektorientierter Planspielentwicklung zu anderen Problemklassen

Planspiele benutzen Simulationen mit realitätsnahen Modellen. Spieler sind Entscheider für Unternehmen, die - wie in der Realität auch - am Markt um Vorteile kämpfen. Was liegt also näher, als zu überprüfen, ob objektorientierte Planspielentwicklung sich auch dazu eignet, Synergien für andere Problemklassen hervorzubringen. Welche Aspekte von Planspielen und Planspielentwicklung bieten sich für Synergieeffekte an und wofür könnten sie genutzt werden? Die folgende Tabelle soll im Sinne möglicher Aspekte verstanden werden:

*Tabelle 24: Bezüge von Konzepten objektorientierter Planspiele zu anderen Anwendungsfällen*

| Aspekt | Bezug |
|---|---|
| Komfortables und flexibles Simulationsmodell mit variablen Auswertungen | Entscheidungsunterstützung auf höher aggregierter Ebene:<br>- Rückversicherungs-Programme<br>- Marktsimulationen<br>- Bilanzsimulationen |
| Objektorientiertes Modell vom Unternehmen | Aufbau eines Unternehmens-Objektmodells als Alternative zu den Unternehmens-Datenmodellen.<br>Unterstützung für flexible, vernetzte Informationssysteme |
| Interaktivität | Verbesserung der Benutzerorientierung durch M-V-C ist keine Utopie |

Die nächstliegende Idee ist die der Nutzung des Simulationsmodells zur Entscheidungsunterstützung. Sicherlich, Simulationsrechnungen werden schon seit langem zur Vorbereitung von Entscheidungen verwendet. Doch dazu ist meist ein gerüttelt Maß an mathematischem Wissen (hauptsächlich werden Gleichungssysteme erstellt) und Programmierkönnen erforderlich. Für eine brauchbare Entscheidungsunterstützung ist es notwendig, daß die Modelle in ihren Grundzügen möglichst den realen Gegebenheiten entsprechen. Hierzu müssen die Wirkrelationen mit Hilfe der Analyse historischer, experimenteller oder subjektiver Daten entsprechend kalibriert werden.[279] Objektorientierte Planspielentwicklungsumgebungen können benutzerfreundliche, anwendungsnahe Werkzeuge anbieten, die auch Nicht-Programmierern den Zugang ermöglichen. Ist das benötigte Wirkungsgefüge mit den vorhandenen Objekten nicht zu realisieren, müssen für diesen Einsatzzweck spezialisierte Klassen implementiert werden. Im Vergleich zur Arbeit mit konventionellen Simulationssystemen ist dies in objektorientierten Systemen durch die Nutzung von Wiederverwendung und Vererbung mit verhältnismäßig geringem Aufwand verbunden. Die Risikoprofile von Versicherungsbeständen können durch die Verfügbarkeit diskreter Verteilungen relativ leicht an reale Gegebenheiten angepaßt werden. Spielen Tarifmerkmale ein Rolle (zum Beispiel bei Marktsimulationen für neue Produkte), so müssen wohl in der Regel Spezialisierungen von Klassen existierender Risikotypen neu entwickelt werden. Ebenfalls geeignet scheint das Simulationsmodell eines objektorientierten Unternehmensplanspiels zur Durchführung von Bilanzsimulationen.

Visionär erscheint ein anderer Bezug, den Konzepte objektorientierter Planspiele aufzeigen: Warum soll man nicht beispielsweise ein Versicherungsunternehmen mit seinen Produkten, Beständen, Kunden, (Vertriebs)-Organisationen im objektorientierten Verständnis modellieren und als Basis für ein vernetztes, unternehmensweites und trotzdem flexibles Informationssystem nutzen? Dem konventionellen Ansatz mit Unternehmensdaten- und Funktionsmodellen als methodische Basis sowie hierarchischen und relationalen Datenbanksystemen zur Implementierung fehlt es an der nötigen Flexibilität[280], deshalb setzen *Frank* / *Klein* die Idee der objektorientierten Unternehmensmodellierung entgegen.[281] Doch diese Ideen sind noch nicht im praktischen Einsatz. Ein Werkzeug ähnlich einer objektorientierten Planspielentwicklungsumgebung könnte die Entscheidung für ein solches System transparenter machen, indem es Konzepte und Anwendungen objektorientierter Unternehmensarchitekturen interaktiv demonstriert.

---

279 Zur Quantifizierung von Reaktionsfunktionen: vgl. Hannsmann, F. (Quantitative Betriebswirtschaftlehere, 1985), S. 115 ff.

280 Dies zeigt sich insbesondere bei der Modellierung von Systemen mit komplexen Datenbeständen und Funktionen (beispielsweise komplexe, veränderliche Anfragen).
Vgl. Picot, A., Maier, M. (Informationsmodellierung, 1994), S. 116

281 *Frank* / *Klein* projektieren am Beispiel der Versicherungsbranche ein objektorientiertes Unternehmensgesamtmodell. Dieses Geamtmodell wird als branchenspezifische, generische Klassenbibliothek implementiert. Ihrer Vision entsprechend, sollen Bibliotheken für andere Branchen folgen.
Vgl. Frank , U., Klein, S. (Unternehmensmodelle, 1992), S. 31 ff.

Gleiches gilt für die Anwendung des M-V-C Ansatzes - Ein Planspielmodell kann mit seinen benutzerorientierten Werkzeugen ein Beispiel für die Implementierung besserer Applikationen werden. Ein Beispiel, das wegen des realitätsnahen Modells eine authentischere Demonstration ermöglicht, als dies konventionelle Prototypen können.

„Phantasie ist wichtiger als Wissen“[282]

## 6.3. Ausblick

Wie hat *Einstein* diesen Ausspruch wohl gemeint? Ein Blick nach vorne kann die Bedeutung dieser Worte für unsere Zeit deutlicher machen: Künftig wird es allein nicht mehr weiterhelfen, das Wissen im Sinne des Humbolt’schen universellen Bildungskonzepts ständig zu vermehren und zu verfeinern - Phantasie und die Fähigkeit, komplexe, schnell veränderliche Situationen zu meistern, werden zunehmend gefordert sein.

### Herausforderung Informationsgesellschaft

Die globale Informationsgesellschaft nimmt allmählich konkrete Züge an.[283] Sichtbare Zeichen sind beispielsweise die gewaltigen Anstrengungen zur Installation eines internationalen Information-Superhighways, der den Transport von Informationen aller Art an jeden gewünschten Ort zu jeder gewünschten Zeit ermöglichen soll.[284] Es entstehen immer neue (elektronische) Wissensreservoirs, die auf diesem Wege zugänglich gemacht werden. Blickt man auf die ungeheuer dynamische und unstrukturierte Entwicklung des Internet[285], kann man sich vorstellen, daß sich die Anforderungen an diejenigen, die von den gespeicherten Informationen profitieren möchten, rapide verändern: Es ist das disziplinen-übergreifende Verständnis von den Aufgaben, das künftig gefordert wird. Neue Mechanismen zur Erschließung der vorhandenen Informationen werden es leichter machen, zu einem Thema problemrelevante Informationen zu bekommen[286];

---

282 Einstein, A.

283 Vgl. Bangemann, M. (Informationsgesellschaft, 1995), S. 1 ff.

284 o.V. (Al Gores Botschaft, 1995), S. 32

285 Das Internet verzeichnet seit 1992 exponentielle Wachstumsraten: Ende 1994 waren etwa 25 Millionen Teilnehmer angeschlossen. Treibende Technologie ist dabei das World Wide Web (kurz: WWW), das, entwicklet vom CERN, ein Hypertextsystem zur verteilten Repräsentation von Informationen bietet.

286 Vgl. Mesaric, G. (Extras, 1995), S. 162 ff.
*Mesaric* stellt das an der Technischen Universität Graz entwickelte Konzept von Hyper-G vor - einer interaktiven Hypermedia-Architektur. Hyper-G bietet einen Ansatz, der die bisher unstrukturiert und chaotisch verlaufende Expansion des Internet mit seiner Vielzahl an Werkzeugen integriert, um interaktive Aspekte ausbaut, das Verknüpfungskonzept erweitert und neue Techniken zur Navigation und Visualisierung verteilter Informationen entwickelt.
Ähnliche, jedoch auf der Präsentationsebene wesentlich weitergehende, Forschungsbemühungen werden schon seit einiger Zeit am PARC mit dem Information Theatre verfolgt.

Problemdefinition, Interpretation und Verständnis aber müssen vom Benutzer mitgebracht werden.

Planspiele versprechen ein geeignetes Instrument zur Lösung einiger in Zukunft immer dringender werdenden Probleme der Aus- und Weiterbildung zu sein. Die Probleme der Anwendbarkeit gelernten Wissens werden ebenso adressiert, wie Planspiele dazu beitragen können, die notwendigen Fähigkeiten zum Umgang mit dem exponentiell wachsenden Informationsangebot - oft als Problem des Information-Overload bezeichnet - aufzubauen.[287]

### Herausforderung Qualität

Die Qualität der erbrachteten Leistungen ganz im Sinne der Erfüllung der Kundenbedürfnisse eines Total Quality Managements (TQM) wird in Zukunft noch mehr in den Mittelpunkt des Interesses rücken und gleichberechtigt neben den bisher dominierenden Kosten-, Mengen- und Terminzielen stehen.[288] Qualitätsorientierung wird den disziplinenübergreifenden Rahmen setzen: Diejenigen, die Leistungen anbieten, werden vor allem Fähigkeiten mitbingen müssen, die sich auf die Kommunikation mit dem Kunden und der Erfassung seiner Bedürfnisse konzentriert. Die verwendeten Methoden und Werkzeuge werden diesem sich vor dieser Forderung messen lassen müssen: Planspiele tragen dazu bei, die notwendige interdisziplinäre System- und Sozialkompentenz hierfür zu vermitteln. Ihre Entwicklung selbst ist dabei gleichzeitig eine Aufgabe, die in das Netzwerk der Kunden/Lieferantenbeziehungen eingebunden ist.

### Herausforderung Planspiel

Mit Planspielen wird der Schwerpunkt des Lernens von der Vermittlung spezialisierten Faktenwissens auf das Erleben und Erlernen von Systemkompetenz verlagert: Sie leisten vor dem Hintergrund der Ideen der Reformpädagogik einen wichtigen Beitrag zur Integration der Disziplinen. Doch die Verbreitung dieser interaktiven Lehrmethode ist trotz steigender Anwendungen noch nicht weit fortgeschritten - es bleibt noch breiter Raum für Verbesserungen. Daraus ergeben sich Ansatzpunkte für weitergehende Arbeiten:

- Die Seminarkonzepte und ihre Integration in bestehende Aus- und Weiterbildungsprogramme müssen weiter verbessert werden: Wenn die Vermittlung theoretischen Wissens und die interaktive Form des Erlebens und Lernens im Planspiel fächerübergreifend koordiniert werden, profitieren am Ende alle Beteiligten.

- Der Ansatz des Cognitive Apprenticeship kann die methodische Basis für diese Konzepte sein. Er fordert jedoch eine verstärkte pädagogische Ausbildung der

287 Vgl. Müller-Zantop, S. (Megatrends, 1995), S. 84

288 Vgl. Crosby, P. (Quality, 1986) S. 3

Lehrenden und einen höheren (Personal)Einsatz während der Durchführung, als dies konventionelle Lehrformen - wie Vorlesungen und Vorträge - tun. Simulationsbasierte Selbstlernumgebungen, basierend auf erweiterten Konzepten dynamischer Planspiele und CBT, können im Gegenzug dazu einen Beitrag leisten, die Vermittlung und Vertiefung von Grundlagenwissen kostengünstiger zu realisieren.[289]

- Eine Umorientierung der Aus- und Weiterbildungskonzepte in Richtung vernetztem Lernen bedingt die disziplinenübergreifende Restrukturierung der Curricula sowie des Prüfungs- und Beurteilungswesens. Dies ist langwieriger Prozeß der Änderung von Konventionen und Traditionen, ohne den allerdings Planspiele nicht aus ihrem Randdasein im Bildungsbereich heraustreten werden.

- Damit sich Planspiele auf ihre eigentliche Aufgabe - dem Lehren und Lernen - konzentrieren können, muß die Technik zur Verfügung stehen, die Lösungen für diese Anforderungen bietet. Die vorliegende Arbeit hat versucht, die Möglichkeiten der objektorientierten Methodik und Softwaretechnologie für diesen Zweck aufzuzeigen. Zunächst müssen praktische Erfahrungen mit der neuen Technologie gesammelt und die Objektbibliotheken zu erweitert und verbessert werden.

## Herausforderung Objekttechnologie

Der objektorientierte Ansatz wird sich in Zukunft eine noch größere Breite von Anwendungen erschließen. Die größe Stärke dieses Ansatzes, seine problemorientierte, verständliche Art und Weise der Modellierung, wird durch die softwaretechnischen Vorteile wie Wiederverwendung, Kapselung und Vererbung unterstützt. In einem Anwendungsfall wie der Planspielentwicklung schließt sich der Kreis: Objektorientierung kann als die Basis für TQM in der Softwareentwicklung dienen[290], Planspielentwicklung kann diese Prinzipien und Konzepte benutzen und das Planspiel selbst trägt dazu bei, die notwendigen Basisfähigkeiten für ein integratives, kundenorientiertes Verständnis zu legen. Denn:

„Bildung ist das, was übrig bleibt,
wenn wir unser Wissen vergessen.“ [291]

289 Vgl. Helten, E. (Vorwort, 1993), S. V

290 Vgl. Wolff, F. (Total Quality, 1994) S. 41 f.

291 Weigele, G. (Führungsthesen, 1992), S. 4

# 1. Literaturverzeichnis


Achtenhagen, F. (Betriebswirtschaftslehreunterricht, 1992) Zum Einsatz von Planspielen im Betriebswirtschaftslehreunterricht, in: Zeitschrift für Planung, 1/1993, S. 3-19

Atwood, T. (Objekt-DBMS-Standard, 1994) Der Objekt-DBMS-Standard, Teil 2, in: OBJEKTspektrum, 2/1994, S. 63-69

Bamberg, G., Coenenberg, A. (Entscheidungslehre, 1981) Betriebliche Entscheidungslehre, 3. Auflage, München 1981

Bangemann, M. (Informationsgesellschaft, 1995) Europas Weg in die Informationsgesellschaft, in: Informatik Spektrum, 18/1995, S. 1-3

Bayerische Rückversicherung AG (Das Planspiel, 1982) Das Planspiel oder Im Sandkasten sieht man klarer, in: Journal Nr. 1 der Bayerischen Rück, 1982

Behne, H. (Paradies, 1994) Künstliches Paradies, in: iX, 2/1994, S. 118-122

von Bertalanffy, L. (Systemtheorie, 1962) Allgemeine Systemtheorie, übersetzt und neu veröffentlicht in: Entscheidungstheorie: Texte und Analysen, Hrsg. Witte, E., Thimm, A., Wiesbaden 1977, S. 235-273

Beyer, T. (Objektbörse, 1993) Objektbörse, in: iX, 2/1993, S. 24-33

Biedenkopf, K. H. (Komplexität, 1994) Komplexität und Kompliziertheit, in: Informatik Spektrum, 1/1994, S. 82-84

Böhret, C., Wordelmann, P. (Fortbildung, 1975) Das Planspiel als Methode der Fortbildung, in: Verwaltung und Fortbildung, Schriftenreihe der Bundesakademie für öffentliche Verwaltung, Köln Bonn 1975

Booch, G. (OOD, 1991) Object Oriented Design with Applications, Redwood City 1991

Broer, H. (Software-Montagetechnik, 1994) Software-Montagetechnik mit Software-Komponenten aus dem Katalog, in: OBJEKTspektrum, 5/1994, S. 68-77

Bronner, R. (Komplexität, 1992) Komplexität, in: Handwörterbuch der Organisation, Hrsg. Frese, E., 3. Auflage, 1992, Sp. 1122-1130

Budde, R. et al (Anwendungssysteme, 1991), Objektorientierter Entwurf benutzerorientierter Anwendungssysteme, in: Software Technik Trends, Band 11, Heft 3, August 1991, S. 184-202

Burger, E. (Communicate, 1994) Communicate it, in: iX, 6/1994, S. 46-52

Buteweg, J. (Systemtheorie, 1988) Systemtheorie und ökonomische Analyse, Pfaffenweiler 1988

Collins, A. (Learning Environments, in Vorbereitung) Design Issues for Learning Environments, in: International perspectives on the psychological foundations of technology-based learning environments, Hrsg.: Vosniadou, S., De Corte, R., Glaser, R., Mandl, H. , New York, in Vorbereitung

Collins, A., Brown, S., Newman, S. (Cognitive Apprenticeship, 1989) Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing and Mathematics, in: Knowing, learning and instruction: Essays in honor of Robert Glaser, Hrsg. Resnick, L., Hillsdale Ney York 1989, S. 453-494

Cox, B. (OOP, 1986) Object Oriented Programming - An Evolutionary Approach, Reading 1986

Crosby, P. (Quality, 1986) Quality without Tears, New York Singapore 1986

Danzer, H. (CBT, 1994) CBT-Entwicklung und -Einsatz in der Hochschulausbildung, in: Information Management, 4/1994, S. 12-19

Dörner, D. (Logik, 1993) Die Logik des Mißlingens, Hamburg 1993

Dier, M., Lautenbacher S. (Groupware, 1994) Groupware - Technologien für die lernende Organisation, München 1994

Digitalk (Smalltalk/V, 1993) Smalltalk/V, Programming Reference, Santa Ana 1993

Drespling, W., Lipps, P. (OpenStep, 1994) OpenStep - Eine objektorientierte Anwendungs- und Entwicklungsumgebung, in: OBJEKTspektrum, 3/1994, S. 26-32

Droege, D. (Objektiv, 1994) Objektiv betrachtet, in: c’t, 11/1994, S. 278-285

Ebert, R. (Lady Ada, 1994) Neues von Lady Ada, in: iX, 10/1994, S. 146-153

Felderer, B., Homburg, S. (Makroökonomik, 1991) Makroökonomik und neuere Makroökonomik, 5. Auflage, Berlin Heidelberg New York 1991

Forrester, J. W. (Dynamics, 1968) Industrial Dynamics, Cambridge, MA, 1968

Forrester, J. W. (Regelkreis, 1971) Der teuflische Regelkreis, Stuttgart, 1971

Frank , U., Klein, S. (Tools, 1992) Three Integrated Tools for Designing and Prototyping Object-Oriented Enterprise Models, Arbeitspapiere der GMD 689, St. Augustin 1992

Frank , U., Klein, S. (Unternehmensmodelle, 1992) Unternehmensmodelle als Basis und Bestandteil integrierter betrieblicher Informationssysteme, Arbeitspapiere der GMD 629, St. Augustin 1992

Frauenstein T., Pape U., Wagner O. (Sprachkonzepte, 1990) Objektorientierte Sprachkonzepte und diskrete Simulation, Berlin Heidelberg 1990

Freiburghaus, M. (Simulation betriebswirtschaftlicher Systeme, 1993) Methodik und Werkzeuge in der Simulation betriebswirtschaftlicher Systeme, St. Gallen 1993

Fricker, U. (System, 1982) Die Versicherungsunternehmung als lebensfähiges System, St. Gallen 1982

Gille, M. (ODMG 93,1994) Die C++ Sprachanbindung im ODMG 93-Standard, in: OBJEKTspektrum Heft 2/1994, 1994, S. 70-73

Graf, J. (Marktübersicht, 1992) Marktübersicht Planspiele, in: Planspiele - Simulierte Realitäten für den Chef von morgen, Hrsg. Graf, J., Bonn 1992, S. 95-250

Graf, J. (Planspiele, 1992) Geleitwort, in: Planspiele - Simulierte Realitäten für den Chef von morgen, Hrsg. Graf, J., Bonn 1992, S. 5-6

Goldberg, A., Robson, D. (Smalltalk, 1989) Smalltalk-80, the Language and its implementation, Reading (MA) 1989

Günther, T. (Management, 1993) Planspiele unterstützen das strategische Management, in: Personalwirtschaft 6/1993, S. 25-28

Haacke, V. (Analyse, 1989) Das „Integrierte Versicherungsunternehmensspiel“ - Analyse des Spielaufbaus, Spielablaufs und betriebswirtschaftlicher Auswirkungen unterschiedliche

Parameterkonstellationen, Diplomarbeit am Institut vom Prof. Dr. B. Kromschröder, Universität Passau, 1989

Hanisch, H.-M. (Petri-Netze, 1992) Petri-Netze in der Verfahrenstechnik, München Wien 1992

Hannsmann, F. (Quantitative Betriebswirtschaftslehre, 1985) Quantitative Betriebswirtschaftslehre, 2. Auflage, München Wien 1995

Harbrücker, U. (Wertewandel, 1992) Wertewandel und Corporate Identity: Perspektiven eines gesellschaftlich orientierten Marketing von Versicherungsunternehmen, Wiesbaden 1992

Hartinger, R. (Puzzle, 1994) Syntax-Puzzle, in: c't, 9/1994, S. 184-188

Häuslein, A., Page B. (DYNAMIS, 1986) DYNAMIS: Ein Modellbildungs- und Simulationssystem mit objektorientierter Benutzeroberfläche, Hamburg 1986

Hajek, F. A. (Theorie komplexer Phänomene, 1972) Die Theorie komplexer Phänomene, Thübingen 1972

Heidack, C. (Lerninstrument, 1992) Lerninstrument an Hochschulen und in der Wirtschaft, in: Planspiele - Simulierte Realitäten für den Chef von morgen, Hrsg. Graf, J., Bonn 1992, S. 45-58

Helten, E. (Prognosemethoden, 1976) Planung betrieblicher Prozesse im VU unter Anwendung von Prognosemethoden, in: VW 8/1976, S. 440-444

Helten, E. ( Schadensummenverteilungen, 1976) ZurTypisierung von Schadensummenverteilungen, in VW 2/1976, S. 113-120

Helten, E. (Konjunktur, 1983) Geamtwirtschaftliches Wachstum und Konjunktur, Versicherungsnachfrage und Marktstrategien in der Schaden- und Unfallversicherung, in: VW 18/1983, S. 1190-1202

Helten, E. (Vorwort, 1993) Vorwort des Herausgebers, in: Schmidt, H. (integratives Konzept, 1993), S. V-VI

Helten, E. (Risiko, 1994) Risiko, Versicherung, Markt, in: Festschrift für Walter Karten zur Vollendung des 60. Lebensjahres, Hrsg. Hesberg, D. et. al., Karlsruhe 1994, S. 19-25

Helten, E., Karten, W. (Kalkulation, 1983) Das Risiko und seine Kalkulation (Teil I), in: Versicherungsenzyklopädie, Hrsg. Große, W. et. al., Band 2: Versicherungsbetriebslehre (VBL), Wiesbaden 1983, S. 3-51

Helten, E., Schmidt, H. („Spiel“ Versicherungen) Das „Spiel“ Versicherung spielend lernen, in: Die Dienstleistung Versicherungsschutz in Wissenschaft und Ausbildung, Sonderdruck, Karlsruhe 1991, S. 80-103

Helten, E., Schmidt, H., Schneider, W. (Qualitätszirkel, 1992) Qualitätszirkel, in: VW, 16/1992, S. 998-1002

Hertz, D. B., Thomas, H. (Risk Analysis, 1984) Practical Risk Analysis - An Approach Through Case Histories, Chichester 1984

Hesse, W., Merbeth, G., Frölich, R. (Software-Entwicklung, 1992) Software-Entwicklung, München Wien 1992

Högsdal, B. (Planspiele, 1992) Die Entwicklung kundenspezifischer Planspiele, in: Planspiele - Simulierte Realitäten für den Chef von morgen, Hrsg. Graf, J., Bonn 1992, S. 83-94

Hruschka, P. (Designprinzipien, 1994) Objektorientierte Analyse- und Designprinzipien, in: OBJEKTspektrum 2/1994, S. 16-22

Hüskes, R. (Geduldsspiel, 1994) Geduldsspiel, in: c't, 9/1994, S. 180-182

Hüskes, R. (Musketiere, 1993) Drei Musketiere, in: c't, 9/1993, S. 174-180

Islinger, G. (Schadensimulation, 1991) Möglichkeiten und Probleme der Schadensimulation - dargestellt an einem Planspiel für Versicherungsunternehmen, Diplomarbeit am INRIVER, Universität München, Prof. Dr. E. Helten, 1991

Itter, F. (Integrierte Modellbildung, 1989) Integrierte Modellbildung und Simulation mit Petrinetzen, in: Zeitschrift für wirtschaftliche Fertigung und Automatisierung, 84/1989, S. 90-92

Johnson, R. E., Foote, B. (Reusable Classes, 1988) Designing Reusable Classes, in: Journal of Object Oriented Programming, 2/1988, S. 22-35

Kavanagh, D. (OMT, 1994) Der OMT-Entwicklungsprozeß im Jahre 1994, in: OBJEKTspektrum, 4/1994, S. 59-65

Kemper, A. Moerkotte, G. (Basiskonzepte, 1993) Basiskonzepte objektorientierter Datenbanksysteme, in: Informatik Spektrum, 16/1993, S. 69-80

Karten, W. (Versicherbarkeit, 1972) Zum Problem der Versicherbarkeit und zur Risikopolitik des Versicherungsunternehmens, in: ZVersWiss 1972, S. 279-299

Kirsch, W. (Betriebswirtschaftslehre, 1993) Betriebswirtschaftslehre, München 1993

Kirsch, W. (Entscheidungsorientierte Betriebswirtschaftslehre, 1989) Entscheidungsorientierte Betriebswirtschaftslehre und und angewandte Forschung, in: Die Betriebswirtschaftslehre im Spannungsfeld zwischen Generalisierung und Spezialisierung, Kirsch, W. , Picot, A., Wiesbaden 1989

Klotzbücher, R. (Aspekte, 1990) Aspekte der Entwicklung wiederverwendbarer Individualsoftware mit objektorientierter Programmierung, Diplomarbeit am Institut für betriebswirtschaftliche Information und Kommunikation, Universität München, Prof. Dr. A. Picot, 1990

Kneer, G., Nassehi, A. (Theorie sozialer Systeme, 1993) Niklas Luhmanns Theorie sozialer Systeme: Eine Einführung, München 1993

Krohn, W., Küppers, G. (Emergenz, 1992) Selbstorganisation. Zum Stand einer Theorie in den Wissenschaften, in: Emergenz: Die Entstehung von Ordnung, Organisation und Bedeutung, Hrsg. Krohn, W., Küppers, G., Frankfurt/M. 1992, S. 7-26

Lamatsch, A (Versicherungsplanspiel, 1984) Das Integrierte Versicherungsplanspiel von Knud Hansen, Studienarbeit am Institut für Wirtschaftstheorie und Operations Research Universität Karlsruhe, Prof. Dr. M. Morlock, 1984

Lehmann, A. (Führungsdimension, 1989) Qualitätsmanagement: Ein bewußte Führungsdimension für den Versicherer?, in: VW 1989, S. 664-673

Lehmann, A. (Dienstleistungsmanagement, 1993) Dienstleistungsmangement: Strategien und Ansatzpunkte zur Schaffung von Servicequalität, Band 9, Reihe Entwicklungstendenzen im Management, ifb-Schriften, Stuttgart, 1993

Letters, F. (Client/Server, 1995) Wie macht der Praktiker 1995 objektorientierte Client/Server-Systeme?, in: OBJEKTspektrum 6/1995, S. 36-39

Liebl, F. (Simulation, 1992) Simulation, München 1992

Mandl, H., Gruber, H., Renkl, A. (Lernen, 1993) Lernen in Schule und Hochschule, Forschungsbericht Nr. 16 des Instituts für Pädagogische Psychologie und Empirische Päadagogik, Universität München, Prof. Dr. H. Mandl, München 1993

Manhart, K. (Komplexität, 1994) Computersimulation dynamischer Systeme: Reduzierte Komplexität, in: MC extra 9/1994, S. 18-25

Meadows, D. H. (Grenzen, 1972) Die Grenzen des Wachstums, Stuttgart 1972

Merten, K. (Kommunikation, 1977) Kommunikation, Opladen 1977

Mesaric, G. (Extras, 1995) Extras inklusive, in: iX 3/95, S. 162 ff.

Meyer, B. (Software Construction, 1988) Object Oriented Software Construction, New York London Toronto 1988

Meyer, M. (Operations Research, 1990) Operations Research - Systemforschung, 3. Auflage, Stuttgart 1990

Michalski, R., Stepp, R. (Observation, 1983) Learning from Observation: Conceptual Clustering, in: Machine Learning: An Artificial Intelligence Approach, Palo Alto 1983, S. 301-352

Mikro Partner (CAPS, 1991) Computer Aided Process Design, Hamburg 1991

Mittendorfer, J. (Vergleich, 1994) Smalltalk - Systeme im Vergleich, in: OBJEKTspektrum Heft 2/1994, S. 45-49

Müller-Zantop, S. (Megatrends, 1995) Die Megatrends der IV, in: Business Computing, 3/1995, S. 84

Naetar, F. (Smalltalk langsam?, 1994) Ist Smalltalk zu langsam?, in: OBJEKTspektrum Heft 3/1994, S. 74-77

o.V. (Al Gores Botschaft, 1995) Industriestaaten nehmen Al Gores Botschaft auf, in: Süddeutsche Zeitung, Nr. 47 vom 25./26.2.1995, S. 32

Pape, U., Schoepf, V., Schwidder, K. (Petri Nets, 1994) Programming by Petri Nets, Informationssschrift des Instituts für Angewandte Informatik Prof. Dr. U. Pape, Technische Universität Berlin, 1994

Picot, A. (Organisation, 1993) Organisation, in: Vahlens Kompendium der Betriebswirtschaftslehre, Hrsg. Bitz, M. et. al., Band 2, 3. Auflage, München 1993, S. 100-174

Picot, A., Franck, E. (Vertikale Integration, 1993) Vertikale Integration, in: Ergebnisse empirischer betriebswirtschaftlicher Forschung: Zu einer Realtheorie der Unternehmung, Festschrift für E. Witte, Hrsg. Hauschildt, J, Grün, O., Stuttgart, 1993, S. 179-219

Picot, A., Gerhardt, T., Nippa, M. (Leistungstiefenentscheidungen, 1992) Strategiekonforme Leistungstiefenentscheidungen, in: Harvard Manager, 3/1992, S. 136-142 (Im Manuskript: S. 1-18)

Picot, A., Maier, M. (Informationsmodellierung, 1994) Ansätze der Informationsmodellierung und ihre Betriebswirtschaftliche Bedeutung, in: zfbf 46. Jg., 2/1994 S. 107-126

Picot, A., Reichwald, R. (Kooperationsformen, 1994) Auflösung der Unternehmung? - Vom Einfluß der IuK-Technik auf Organisationsstrukturen und Kooperationsformen, in: ZfB, 64. Jahrgang, Heft 5, 1994, S. 547-570 (Im Manuskript: S. 1-24)

Pountain, D. (Oberon/F, 1995) The Oberon/F System, in: BYTE January 1995, S. 227-228

Puder, A. (Objekt, 1994) Objekt im Objekt im Objekt, in: iX, Heft 6/1994, S. 54-60

Quibeldey-Cirkel, K. (Paradigmenwechsel, 1994) Paradigmenwechsel im Software-Engineering: Auf dem Weg zu objektorientierten Weltmodellen, in: Softwaretechnik-Trends, Februar 1994, S. 47-58

Rapoport, A. (Systemtheorie, 1988) Allgemeine Systemtheorie, Darmstadt, 1988

Reimann-Rothmeier, G., Mandl, H. (Wissensvermittlung, 1994) Wissensvermittlung: Ansätze zur Förderung des Wissenserwerbs, Forschungsbericht Nr. 34 des Instituts für Pädagogische Psychologie und Empirische Pädagogik, Universität München, Prof. Dr. Heinz Mandl, München 1994

Rumbaugh, J. (MVC, 1994) Eine Betrachtung der Architektur Model-View-Controller (MVC), in: OBJEKTspektrum Heft 3/1994, S. 49-54

Rohn, W. (Führungsentscheidung, 1964) Führungsentscheidung im Unternehmensplanspiel, Essen 1964

Rohn, W. (Didaktik, 1980) Methodik und Didaktik des Planspiels, Köln 1980

Rohn, W. (Planspiel-Übersicht, 1992) Europäische Planspiel-Übersicht 1992, Wuppertal 1992

Rohn, W. (Praxis, 1992) Simulation - Praxis am Modell erlernen, in: Planspiele - Simulierte Realitäten für den Chef von morgen, S. 19-28, Hrsg. Graf, J., Bonn 1992

Ropohl, G. (Systemtheorie der Technik, 1979) Eine Systemtheorie der Technik, München 1979

Rosenbeck, P. (Twelve Years, 1992) Twelve Years After, in: c't, Heft 2/1992, S. 153-157

von Rosenstiel, L. (Organisationspsychologie, 1987) Grundlagen der Organisationspsychologie, 2. Auflage, Stuttgart 1987

De Rosnay, (Makroskop, 1977) Das Makroskop, Stuttgart 1977

Schlageter, G., Stucky, W. (Datenbanksysteme, 1983) Datenbanksysteme: Konzepte und Modelle, Stuttgart 1983

Schwertfeger, B. (Fehler, 1992) Fehler erwünscht, in: Wirtschaftswoche Nr. 42 vom 9.10.1992, S. 69-72

Schmidt, H. (integratives Konzept, 1993) Ein integratives Konzept zur Erstellung von CBT-Programmen, München 1993

Schmidt, H. (Schlüsselqualifikation, 1994) Schlüsselqualifikation ‘Soziale Kompetenz’ im Unternehmensspiel ‘Versicherungen’ , Vortragsunterlagen zum Expertengespräch beim Institut der Deutschen Wirtschaft über Schlüsselqualifikationen im Betriebswirtschaftsstudium, 12.4.1994, Köln

Schmidt, H., Hardt, M. (Strukturwandel, 1994) Strukturwandel - und dann?, in: VW, 21/1994, S. 1433-1435

Schmidt, J. W. (language constructs, 1977) Some high level language constructs for data and type relation, in: ACM Transactions on Database Systems Bd. 2, 1977, S. 247-264

Schmitz, N., Lehmann, F. (Monte-Carlo, 1976) Monte-Carlo-Methoden I - Erzeugen und Testen von Zufallszahlen, Meisenheim, 1976

Schneider, D. (Allgemeine Betriebswirtschaftslehre, 1987) Allgemeine Betriebswirtschaftslehre, 3. Auflage, München 1987

Skudelny, H. (Objektorientierung, 1995) Objektorientierung gewinnt an Boden, in: Office Management 1-2/1995, S. 62-63

Stauss, B., Hentschel, B. (Dienstleistungsqualität, 1991) Dienstleistungsqualität, in: Wirtschaftwissenschaftliches Studium, 5/1991, S. 238-243

Stein, W. (Vergleich, 1993) Objektorientierte Analysemethoden - ein Vergleich, in: Informatik-Spektrum, 16/1993, S. 317-332

Stroustrup, B. (C++, 1992) Die C++ Programmiersprache, 2. Auflage, 4. überarbeiter Nachdruck 1994, Bonn München Paris 1992

Simon, K.-H., (Realität spielen, 1992) Realität spielen, in: c't Heft 9/1992, S. 88-93

Thelen, B. (Ordnung, 1994) Ordnung im Chaos: OO-Methoden, ihre Ursprünge und ein pragmatisches Konzept, in: OBJECTspektrum, 5/1994, S. 78-82

Tilly, T. (Fuzzy-Logik, 1991) Fuzzy-Logik: Grundlagen, Anwendungen, Hard- und Software, München 1991

Unseld, S. (Künstliche Intelligenz, 1990) Künstliche Intelligenz und Simulation in der Unternehmung, Stuttgat 1990

Versteegen, G. (Hälse, 1995) Schlanke Hälse, in iX, S. 86-92

Vester, F. (Neuland, 1991) Neuland des Denkens: Vom technokratischen zum kybernetischen Zeitalter, 7. Auflage, München 1991

Vester, F. (Denken, 1978) Denken, Lernen, Vergessen, 1. Auflage, München 1978

Vorwerk, R. (ODBMS, 1993) ODBMS- Benutzerhandbuch, Braunschweig 1993

Wagner, M. (Objektorientierung, 1995) Welle der Objektorientierung erreicht jetzt auch den deutschsprachigen Raum, in: Computer Zeitung Nr. 7 vom 16. Februar 1995, S. 6

Warth, W. P. (Think, 1992) THINK, in: Impuls, Hauszeitung der Schweizerischen Rückversicherungsgesellschaft 4/1992, Zürich, S.3

Weber, R. (SQL3, 1993) SQL2-Norm und SQL3-Projekt, in: Informatik Spektrum, 16/1993, S. 95

Weigele, G. (Führungsthesen, 1992) Führungsthesen, in: Service als Wettbewerbsvorteil auf dem europäischen Binnenmarkt; Vortragsunterlagen des Europäischen Zentrums für die Bildung im Versicherungswesen zum Seminar vom 7./8. Mai 1992

Werner, U. (VersPlan, 1991) VersPlan PC 1.0, Das integrierte Versicherungsplanspiel von Knud Hansen und INRIVER, Teilnehmerhandbuch, INRIVER Universität München, Prof. Dr. E. Helten, 1991

Wolff, F. (Total Quality, 1994), Ein Total Quality Modell für die Softwareentwicklung, in: Software Technik Trends, Band 14, Heft 1, Februar 1994, S. 38-46

Zerbe, K. (Plaudertasche, 1992) Plaudertasche, in: c’t, Heft 7/1992, S. 110-112

# 1. Autorenverzeichnis


Achtenhagen, F.
Atwood, T.
Bamberg, G.
Bangemann, M.
Behne, H.
Beyer, T.
Biedenkopf, K. H.
Böhret, C.
Booch, G.
Broer, H.
Bronner, R.
Brown, S.
Budde, R.
Burger, E.
Buteweg, J.
Coenenberg, A.
Collins, A.,
Cox, B.
Crosby, P.
Danzer, H.
De Rosnay
Dier, M.,
Dörner, D.
Drespling, W.
Droege, D.
Ebert, R.
Felderer, B.
Foote, B.
Forrester, J. W.
Franck, E.
Frank , U.
Frauenstein T.
Freiburghaus, M.
Fricker, U.
Frölich, R.
Gerhardt, T.
Gille, M.
Goldberg, A.
Graf, J.
Gruber, H.
Günther, T.
Haacke, V.
Hajek, F. A.
Hanisch, H.-M.
Hannsmann, F.
Harbrücker, U.
Hardt, M.
Hartinger, R.
Häuslein, A.
Heidack, C.
Helten, E.
Hentschel, B.
Hertz, D. B.
Hesse, W.
Högsdal, B.
Homburg, S.

Hruschka, P.
Hüskes, R.
Islinger, G.
Itter, F.
Johnson, R. E.
Karten, W.
Kavanagh, D.
Kemper, A.
Kirsch, W.
Klein, S.
Klotzbücher, R.
Kneer, G.
Krohn, W.
Küppers, G.
Lamatsch, A.
Lautenbacher S.
Lehmann, A.
Lehmann, F.
Letters, F.
Liebl, F.
Lipps, P.
Maier, M.
Mandl, H.
Manhart, K.
Meadows, D. H.
Merbeth, G.
Merten, K.
Mesaric, G.
Meyer, B.
Meyer, M.
Michalski, R.
Mittendorfer, J.
Moerkotte, G.
Müller-Zantop, S.
Naetar, F.
Nassehi, A.
Newman, S.
Nippa, M.
Page, B.
Pape, U.
Picot, A.
Pountain, D.
Puder, A.
Quibeldey-Cirkel, K.
Rapoport, A.
Reichwald, R.
Reimann-Rothmeier, G.
Renkl, A.
Robson, D.
Rohn, W.
Ropohl, G.
Rosenbeck, P.
Rumbaugh, J.
Schlageter, G.
Schmidt, H.
Schmidt, J. W.
Schmitz, N.
Schneider, D.
Schneider, W.
Schoepf, V.

Schwertfeger, B.
Schwidder, K.
Simon, K.-H.,
Skudelny, H.
Stauss, B.
Stein, W.
Stepp, R.
Stroustrup, B.
Stucky, W.
Thelen, B.
Thomas, H.
Tilly, T.
Unseld, S.
Versteegen, G.
Vester, F.
von Bertalanffy, L.
von Rosenstiel, L.
Vorwerk, R.
Wagner O.
Wagner, M.
Warth, W. P.
Weber, R.
Weigele, G.
Werner, U.
Wolff, F.
Wordelmann, P.
Zerbe, K.

# 1. Stichwortverzeichis


**—C—**

Cognitive Apprenticeship, 5; 35; 37; 105; 175; 177

**—I—**

INRIVER, XI; 1; 44; 180; 186

Integriertes Versicherungsplanspiel, 1

**—K—**

Komplexität, V; 6; 12; 20; 38; 45; 50; 52; 53; 57; 62; 64; 79; 80; 81; 100; 101; 102; 103; 104; 105; 110; 158; 165; 167; 168; 169; 176; 182

Komplexitätsvariation, 92; 102
- horizontale -, 102
- vertikale -, 102

Kompliziertheit, 6; 12; 92; 95; 96; 127; 157; 167; 176

**—L—**

Lernziele, V; 26; 36; 39
- Sozialkompetenz, 110
- Systemkompetenz, 35; 110

**—M—**

Mikro/Makro-Problem, 98; 100; 101; 102

Mikrowelt, 10

Model-View-Controller, 71; 92; 94; 121; 133

**—O—**

Objektorientierte Datenbanken
- Datenbanksprache, 75
- Evolutionärer Ansatz, 78
- $NF^2$, 78
- Persistenz, 75
- Revolutionärer Ansatz, 74
- SQL-3, 78
- Transaktionskonzept, 76
- VC ODBMS, 77; 130

Objektorientierte Programmierung
- ADA, 73
- C++, 72
- CORBA, 74
- Eiffel, 71
- Hybride Programmiersprachen, 71
- Oberon, 71
- Objective-C, 73
- Rein objektorientierte Programmiersprachen, 69
- Smalltalk, 70; 84; 94; 97; 128
- SOM und DSOM, 73

Objektorientierung
- Datenbanken, 74
- Objektbibliothek, 4; 91; 96; 135; 171
- Objektorientiertes Paradigma, 3

**—P—**

Pädagogik
- Reformpädagogik, V; 2; 34; 174

Petrinetz, 81

Planspielseminar
- Klassische Seminarsituation, 45
- Lehr-/Lern-situationen, 2; 5; 35; 46; 49; 50; 51
- Seminarsituation, 5; 44; 51; 53; 57; 58; 79; 84; 90; 101; 131; 144; 157; 169

**—S—**

Simulationsmodell, 2; 21; 28; 55; 79; 83; 84; 96; 108; 119
- Aufbauorganisation, 92; 97; 102; 103; 135; 147
- Metamodell, 64; 120; 129; 134; 135; 137; 140; 143

Systemtheorie, 2; 10; 14

# 1. Lebenslauf

| | | |
|---|---|---|
| 30.6.1964 | geboren | in München |
| 1970 - 1974 | Schule | Grundschule Unterhaching |
| 1974 - 1983 | | Gymnasium Unterhaching |
| 1983 | | Abitur |
| 1983 - 1985 | Berufsausbildung | zum Industriekaufmann<br>bei der Siemens AG, München,<br>Unternehmensbereich Bauelemente |
| 1985 - 1986 | Wehrdienst | beim Materialkontrollzentrum der Luftwaffe in Erding |
| WS 1986 | Studium | Statistik, an der Ludwig-Maximilians-Universität München |
| SS 1987 - SS 1991 | | Betriebswirtschaftslehre, an der Ludwig-Maximilians-Universität München |
| 1987 - 1990 | Studienbegleitende Tätigkeiten | Werkstudent bei der Siemens AG, München<br>- Zentralbereiche Rechnungswesen und Unternehmensführung<br>- Anwendungsprogrammierung für Expertensysteme |
| WS 1988 - SS 1991 | | Wissenschaftliche Hilfskraft am Institut für Risikoforschung und Versicherungswirtschaft, Prof. Dr. E. Helten, Universität München<br>Durchführung eines größeren Softwareprojekts |
| 1992 - 1995 | Promotion | „Objektorientierte Planspielentwicklung“ am Institut für Risikoforschung und Versicherungswirtschaft,<br>Prof. Dr. E. Helten, Universität München |
| seit SS 1992 | Lehrauftrag | am Institut für Risikoforschung und Versicherungswirtschaft, Prof. Dr. E. Helten, Universität München |
| seit 1991 | | Systemberater im Bereich Organisation / Information bei der Siemens AG, München, Unternehmensbereich Halbleiter |
| seit 1991 | freiberufliche Tätigkeit | als Dozent für die Weiterbildung in der Versicherungswirtschaft |